Solaris 9 9/04 Installation Guide

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

Part No: 817–5768–10
September 2004

040609@9061

Contents

Preface

This book describes how to install and upgrade the Solaris™ 9 operating environment on both networked and non-networked SPARC® and x86 architecture based systems.

This book does not include instructions about how to set up system hardware or other peripherals.

Note – The Solaris operating environment runs on two types of hardware, or platforms—SPARC and x86. The information in this document pertains to both platforms unless called out in a special chapter, section, note, bulleted item, figure, table, example, or code example.

Note – In this document the term "x86" refers to the Intel 32-bit family of microprocessor chips and compatible microprocessor chips made by AMD.

Who Should Use This Book

This book is intended for system administrators responsible for installing the Solaris operating environment. This book provides both of the following types of information.

- Advanced Solaris installation information for enterprise system administrators who manage multiple Solaris machines in a networked environment
- Basic Solaris installation information for system administrators who perform infrequent Solaris installations or upgrades

Related Books

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

TABLE P–1 Related Information

Information	Description
System Administration Guide: Devices and File Systems	Describes how to back up system files
Solaris 9 9/04 Release Notes	Describes any bugs, known problems, software that is being discontinued, and patches that are related to the Solaris release
SPARC: *Solaris 9 Sun Hardware Platform Guide*	Contains information about supported hardware
Solaris 9 9/04 Package List	Lists and describes the packages in the Solaris 9 9/04 operating environment
x86: *Solaris (x86 Platform Edition) Hardware Compatibility List*	Contains supported hardware information and device configuration

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see "Buy printed documentation" at `http://docs.sun.com`.

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–2 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	The command to remove a file is `rm` *`filename`*.
AaBbCc123	Book titles, new terms or terms to be emphasized	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–3 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

CHAPTER 1

Planning for Solaris Installation or Upgrade (Topics)

This section guides you through planning the installation or upgrade of the Solaris operating environment.

Chapter 2	Provides information about decisions that you need to make before you install or upgrade.
Chapter 3	Provides details about the different Solaris installation technologies to help you choose which method is best for your environment.

CHAPTER 2

Planning for a Solaris Installation or Upgrade (Overview)

This chapter provides you with information about decisions you need to make before you install or upgrade the Solaris operating environment. This chapter contains the following sections:

- "Task Map: Installing or Upgrading the Solaris Software" on page 27
- "Initial Installation or Upgrade?" on page 29
- "Installing From the Network or From DVD or CDs?" on page 31
- "Using DVD Media" on page 32
- "Sun ONE Application Server Postinstallation Configuration" on page 32
- "x86: Accessing the Solaris 9 Device Configuration Assistant and PXE" on page 32

Note – This book uses the term *slice*, but some Solaris documentation and programs might refer to a slice as a partition. To avoid confusion, this book distinguishes between `fdisk` partitions (which are supported only in Solaris *x86 Platform Edition*) and the divisions within the Solaris `fdisk` partition, which might be called slices or partitions.

Task Map: Installing or Upgrading the Solaris Software

The following task map is an overview of the steps necessary to install or upgrade the Solaris operating environment. Use this task map to identify all of the decisions that you need to make to complete the most efficient installation for your environment.

TABLE 2–1 Task Map: Installing or Upgrading the Solaris Software

Task	Description	For Instructions
Choose initial installation or upgrade.	Decide if you want to perform an initial installation or an upgrade.	"Initial Installation or Upgrade?" on page 29
Choose an installation method.	The Solaris operating environment provides several methods for installation or upgrade. Choose the installation method that is most appropriate for your environment.	Chapter 3
Review system requirements. Also, plan and allocate disk space and swap space.	Determine if your system meets the minimum requirements to install or upgrade. Allocate disk space on your system for the components of the Solaris operating environment that you want to install. Determine the appropriate swap space layout for your system.	Chapter 5
Choose an installation location.	You can install the Solaris software from local media or from the network. Decide on an installation location that is most appropriate for your environment.	"Installing From the Network or From DVD or CDs?" on page 31
Gather information about your system.	Use the checklist and complete the worksheet to collect all of the information that you need to install or upgrade.	Chapter 6
(Optional) Preconfigure system information.	You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.	Chapter 7
(Optional) Prepare to install the Solaris software from the network.	If you chose to install the Solaris software from the network, create an install server, create a boot server (if necessary), and set up the systems to be installed from the network.	To install over a local area network, see Chapter 15. To install over a wide area network, see Chapter 43.
(Upgrade only) Perform the pre-upgrade tasks.	Back up your system, determine if you can upgrade with disk space reallocation, and search for patches that a Solaris Update release might override.	Chapter 8
Install or upgrade.	Use the Solaris installation method that you chose to install or upgrade the Solaris software.	The chapter or chapters that provide detailed instructions for the installation method
Troubleshooting installation problems	Review the troubleshooting information when you encounter problems with your installation.	Appendix E

Initial Installation or Upgrade?

You can choose to perform an initial installation or, if your system is already running the Solaris operating environment, you can upgrade your system.

Initial Installation

An initial installation overwrites the system's disk with the new version of the Solaris operating environment. If your system is not running the Solaris operating environment, you must perform an initial installation.

If the system is already running the Solaris operating environment, you can choose to perform an initial installation. If you want to preserve any local modifications, before you install, you must back up the local modifications. After you complete the installation, you can restore the local modifications.

You can use any of the Solaris installation methods to perform an initial installation. For detailed information about the different Solaris installation methods, refer to Chapter 3.

Upgrade

An upgrade merges the new version of the Solaris operating environment with the existing files on the system's disk. An upgrade saves as many modifications as possible that you have made to the previous version of the Solaris operating environment.

You can upgrade any system that is running the Solaris 2.6, Solaris 7, or Solaris 8 software. Type the following command to see the version of Solaris software that is running on your system:

```
$ uname -a
```

You can upgrade the Solaris operating environment by using the following installation methods.

Note – Use the `smosservice patch` to upgrade diskless clients. For detailed instructions, refer to *System Administration Guide: Basic Administration* or to `smosservice`(1M).

Solaris Upgrade Methods

TABLE 2–2 SPARC: Solaris Upgrade Methods

Current Solaris Operating Environment	Solaris Upgrade Methods
Solaris 2.6, Solaris 7, Solaris 8, Solaris 9	■ Solaris™ Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart™ method ■ Solaris Live Upgrade

TABLE 2–3 x86: Solaris Upgrade Methods

Current Solaris Operating Environment	Solaris Upgrade Methods
Solaris 2.6	Installing from DVD media or a net installation image: ■ Solaris Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart method Installing from CD media: ■ Solaris `suninstall` program ■ Custom JumpStart method
Solaris 7	Installing from DVD media or a net installation image: ■ Solaris Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart method ■ Solaris Live Upgrade Installing from CD media: ■ Solaris `suninstall` program ■ Custom JumpStart method ■ Solaris Live Upgrade
Solaris 8, Solaris 9	Installing from DVD or CD media or a net installation image: ■ Solaris Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart method ■ Solaris Live Upgrade

Upgrade Limitations

Note – For limitations on upgrading using Solaris Live Upgrade, see “Solaris Live Upgrade Requirements” on page 395.

When upgrading, pay attention to the following issues.

- Upgrading to another software group. You cannot upgrade your system to a software group that is not installed on the system. For example, if you previously installed the End User Solaris Software Group on your system, you cannot use the upgrade option to upgrade to the Developer Solaris Software Group. However, during the upgrade you can add software to the system that is not part of the currently installed software group.
- Upgrading to a Solaris Update release. If you are already running the Solaris 9 operating environment and have installed individual patches, upgrading to a Solaris 9 Update release causes the following:
  - Any patches that were supplied as part of the Solaris 9 Update release are reapplied to your system. You cannot back out these patches.
  - Any patches that were previously installed on your system and are not included in the Solaris 9 Update release are removed.

  You can use the Patch Analyzer to determine which patches, if any, will be removed by upgrading to the Solaris 9 Update release. For detailed instructions about using the Patch Analyzer, refer to “Upgrading to a Solaris Update Release” on page 661.

Installing From the Network or From DVD or CDs?

The Solaris software is distributed on DVD or CD media so that you can install or upgrade systems that have access to a DVD-ROM or CD-ROM drive.

If you have systems that do not have local DVD-ROM or CD-ROM drives or if you are installing several systems and do not want to insert the discs into every local drive to install the Solaris software, you can set up the systems to install from remote DVD or CD images.

You can use all of the Solaris installation methods to install a system from the network. However, by installing systems from the network with the Solaris Flash installation feature or with a custom JumpStart installation, you can centralize and automate the installation process in a large enterprise. For more details about the different installation methods, refer to Chapter 3.

Installing the Solaris software from the network requires initial setup. For information about preparing to install from the network, choose one of the following options.

- For detailed instructions on preparing to install from a local area network, refer to Chapter 15.
- For instructions on preparing to install over a wide area network, see Chapter 43.

- For instructions about how to install x86–based clients over the network by using PXE, see "x86: Booting and Installing Over the Network With PXE" on page 667.

Using DVD Media

When you are using DVD media and are asked to boot from the `ok` prompt, always type the following command:

```
ok boot cdrom
```

Sun ONE Application Server Postinstallation Configuration

If you are using the Sun™ ONE Application Server, you need to configure the server after installing the Solaris operating environment. For information on configuring the Application Server, see *Sun ONE Application Server 7 Standard Edition Getting Started Guide*.

x86: Accessing the Solaris 9 Device Configuration Assistant and PXE

The Solaris Device Configuration Assistant is a program that enables you to perform various hardware configuration and booting tasks. You use the Device Configuration Assistant to boot from either a DVD, a CD, a net installation image, or a copy of the software on a diskette. You can access the Solaris Device Configuration Assistant in the following ways:

- By booting from the Solaris DVD, Solaris Installation CD, or Solaris Software 1 of 2 CD. Your system's BIOS must support booting from a DVD or CD.
- By booting from a net installation image of Solaris Software with Preboot Execution Environment (PXE). PXE enables you to boot a system directly from the network without using the boot diskette. The system must support PXE. Enable the system to use PXE by using the system's BIOS setup tool or the network adapter's

configuration setup Tool. For more information on configuring your system for using PXE network boot, see “x86: Booting and Installing Over the Network With PXE” on page 667.

- By booting from a diskette. You can copy the Device Configuration Assistant software to a diskette from the Solaris *x86 Platform Edition* DVD or Solaris Software 2 of 2 *x86 Platform Edition* CD by using the copy procedure in Appendix I.

CHAPTER **3**

Choosing a Solaris Installation Method (Overview)

This chapter describes the different installation technologies. The Solaris operating environment provides several methods for installation or upgrade. Each installation technology offers different features that are designed for specific installation requirements and environments. Choose the technology that is most appropriate for your environment.

Solaris Web Start Installation Program

The Solaris Web Start installation program on the Solaris DVD or Solaris Installation CD can be run with a graphical user interface (GUI) or with a command–line interface (CLI). The Solaris Web Start program guides you step-by-step through installing or upgrading the Solaris software and additional software. You can install with the default option, or you can use the customize option to install only the software you want.

If you are new to the Solaris operating environment or to UNIX®, the Solaris Web Start program enables you to easily move forward and back during the installation to make changes as necessary. Installation tasks are divided into panels that offer default values for you to select.

Because the Solaris Web Start program prompts you to enter information, you have to interact with the installation program. Consequently, this installation method might not be the most efficient method when you have to install or upgrade several systems. For batch installations of a large number of systems, use custom JumpStart or the Solaris Flash installation feature.

For detailed instructions, refer to Chapter 17.

Solaris `suninstall` Program

The Solaris `suninstall` program on the Solaris Software 1 of 2 CD is run with a command-line interface (CLI). The Solaris `suninstall` program guides you step-by-step through installing or upgrading to the Solaris 9 software. This installation method is good if you have insufficient memory to run a GUI and are running international locales.

The Solaris `suninstall` program only installs the Solaris operating environment software. This program does not know of third–party applications or network-downloadable software. You must install third–party applications after you have installed the Solaris operating environment. Also, you are prompted to enter system configuration information during installation, so `suninstall` is not the most efficient installation method for installing several systems. The Solaris Web Start program enables you to install third-party applications. Or, for batch installations of a large number of systems, use custom JumpStart or the Solaris Flash installation feature.

For detailed instructions, refer to Chapter 18.

Custom JumpStart Installation Method

The custom JumpStart installation method is a command–line interface that enables you to automatically install or upgrade several systems, based on profiles that you create. The profiles define specific software installation requirements. You can also incorporate shell scripts to include preinstallation and postinstallation tasks. You choose which profile and scripts to use for installation or upgrade. The custom JumpStart installation method then installs or upgrades the system.

If you are familiar with the Solaris operating environment and have multiple systems to install, the custom JumpStart installation method might be the most efficient way for you to install your systems. If you plan to install only a few systems, this installation method is less efficient. The creation of a custom JumpStart environment might be time consuming. If you are unfamiliar with the operating environment or have only a few systems to install, use the Solaris Web Start installation program.

For detailed instructions, refer to Chapter 26.

Solaris Flash Installation Feature

The Solaris Flash installation feature allows you to install many systems, based on a configuration that you install on a master system. After you install and configure the master system, you create a Solaris Flash archive from the master system. You can create as many Solaris Flash archives as you need. You choose which Solaris Flash archive to install on each different system. This installation method enables you to efficiently install many systems with the same software and configuration.

When you use any of the Solaris installation methods and you do not select to install a Solaris Flash archive, the installation method installs each Solaris package individually. The package-based installation method is time consuming because the installation method must update the package map for each package. Solaris Flash archives install on your system much faster than when you install each of the individual Solaris packages.

Each of the Solaris installation methods enables you to replicate an initial installation of a Solaris Flash archive. The Solaris Web Start installation method and the Solaris `suninstall` program prompt you to select the Solaris Flash archive that you want to install. If you are performing a custom JumpStart installation, you specify the Solaris Flash archive that you want to install in the profile file. If you are performing a Solaris Live Upgrade, you specify the Solaris Flash archive that you want to install on the inactive boot environment.

If you want to update a previously installed clone system, you can install small changes with a differential archive. The differential archive installs archives that contain only the differences between two system images. You can install a differential archive by using the custom JumpStart installation method or Solaris Live Upgrade.

If you have many different configurations that you want to install on your systems, you need a Solaris Flash archive for each system. Solaris Flash archives are large files and require a significant amount of disk space. If you have many different installation configurations or if you want the flexibility to change your installation configuration, you might consider using the custom JumpStart installation method. Alternatively, you can accomplish system-specific customizations by using a JumpStart finish script or an embedded Flash postdeployment script.

For an overview of Solaris Flash installation features, including installing an archive with an initial installation or an update and using scripts, see Chapter 20.

For detailed instructions on installing an archive with custom JumpStart, see "To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation" on page 317.

WAN Boot Installation Method

The WAN boot installation method enables you to boot and install software over a wide area network (WAN) by using the Hypertext Transfer Protocol (HTTP). You can use WAN boot to install the Solaris operating environment on systems over the Internet or other large public data networks whose network infrastructure might be untrustworthy. You can use different security features to protect data confidentiality and installation image integrity.

The WAN boot installation method enables you to transmit an encrypted Solaris Flash archive over a public network and perform a custom JumpStart installation on a remote client. To protect the integrity of the installation, you can use private keys to authenticate and encrypt data. You can also transmit your installation data and files over a secure HTTP connection by configuring your systems to use digital certificates.

For overview information and detailed instructions about the WAN boot installation method, see Chapter 40.

Solaris Live Upgrade Method

Upgrading with Solaris Live Upgrade enables a duplicate boot environment to be upgraded while the active boot environment is still running, thus eliminating downtime of the production environment. Solaris Live Upgrade can be run with a CUI or CLI. You first create a duplicate boot environment. After a duplicate boot environment is created, you can upgrade the boot environment. You can also install a Solaris Flash archive or differential archive on the inactive boot environment. When you are ready, you activate the inactive boot environment and at the next reboot, the inactive boot environment is switched to become the active boot environment. If a failure occurs, you can recover your original boot environment with a simple activate and reboot.

For a detailed overview and instructions, refer to Chapter 32.

SPARC: Factory JumpStart Installation Method

The JumpStart installation method automatically installs the Solaris software on a new SPARC system when you insert the Solaris DVD or Solaris Software 1 of 2 CD into the drive and turn on the system. A default profile is selected that is based on the model and disk size of the system. The profile determines which software components are installed on the system. You are not prompted for system configuration information and you cannot choose which software is installed.

A JumpStart boot image, which is required to use this installation method, is preinstalled on all new SPARC based systems. If you have an older SPARC based system, you can add the JumpStart installation method to the system by using the `re-preinstall`(1M) command. You cannot use the JumpStart installation method on x86 based systems.

CHAPTER 4

Preparing for Solaris Installation or Upgrade (Topics)

This section provides instructions for preparing to install or upgrade the Solaris operating environment.

Chapter 5	Provides guidelines to help you plan the disk space that you need to install or upgrade the Solaris operating environment.
Chapter 6	Provides a comprehensive list of the information about your system that you need to obtain before you begin the Solaris installation.
Chapter 7	Provides instructions for using the `sysidcfg` file to specify system information during the Solaris installation.
Chapter 8	Provides detailed requirements for upgrading the Solaris operating environment.

CHAPTER 5

System Requirements and Guidelines (Planning)

This chapter describes system requirements to install or upgrade to the Solaris operating environment. General guidelines for planning the disk space and default swap space allocation are also provided. This chapter contains the following sections:

- "Memory Requirements" on page 43
- "Allocating Disk Space" on page 44
- "SPARC: How Swap Space Is Allocated" on page 46
- "Requirements When Using the Solaris Installation CD" on page 46
- "x86: Change in Default Boot-Disk Partition Layout" on page 48

Memory Requirements

To install or upgrade to the Solaris operating environment, the suggested memory size is 128 Mbytes or greater.

Note – Some optional installation features are enabled only when sufficient memory is present. For example, if you install from a DVD with insufficient memory, you install through the Solaris Web Start installation program's command-line interface, not through the Web Start graphical user interface.

Allocating Disk Space

Before you install the Solaris 9 software, you can determine if your system has enough disk space by doing some high-level planning. If you plan your disk space needs before you install, you can add more disks to your system, if you need them, before you install the Solaris 9 software. If you are upgrading, review "Upgrading With Disk Space Reallocation" on page 100.

General Disk Space Planning and Recommendations

Planning disk space is different for everyone. Consider the following general points when you are planning disk space.

- Allocate additional disk space for each language you select to install, for example, Chinese, Japanese, Korean.
- If you intend to support printing or mail, allocate additional disk space in the `/var` file system.
- If you intend to use the crash dump feature `savecore`(1M), allocate double the amount of your physical memory in the `/var` file system.
- If a server provides home directory file systems for users on other systems, allocate additional disk space on the server. By default, home directories are usually located in the `/export` file system.
- Allocate at least 512 Mbytes of swap space.
- Allocate space for the Solaris software group you want to install. The recommended disk space for the software groups is in Table 5–1. When you are planning disk space, remember that you can add or remove individual software packages from the software group that you select.
- Create a minimum number of file systems. By default, the Solaris installation methods create only root (`/`) and `/swap`. When space is allocated for OS services, the `/export` directory is also created. For each file system that you create, allocate an additional 30 percent more disk space than you need to enable you to upgrade to future Solaris versions. Each new Solaris release needs approximately 10 percent more disk space than the previous release. By allocating an additional 30 percent of disk space for each file system, you allow for several Solaris upgrades before you need to reslice your system.
- Allocate additional disk space for additional software or third-party software.

If you are using Solaris Live Upgrade, see "Solaris Live Upgrade Disk Space Requirements" on page 396.

Disk Space Recommendations for Software Groups

The Solaris software groups are collections of Solaris packages. Each software group includes support for different functions and hardware drivers. You select the software group to install, based on the functions that you want to perform on the system.

- Core System Support Software Group – Contains the minimum code that is required to boot and run a networked Solaris system.
- End User Solaris Software Group – Contains the minimum code that is required to boot and run a networked Solaris system and the Common Desktop Environment.
- Developer Solaris Software Group – Contains the End User Software Group plus additional support for software development. The additional software development support includes libraries, include files, man pages, and programming tools. Compilers are not included.
- Entire Solaris Software Group – Contains the Developer Solaris Software Group and additional software that is needed for servers.
- Entire Solaris Software Group Plus OEM Support – Contains the Entire Solaris Software Group plus additional hardware drivers, including drivers for hardware that is not on the system at the time of installation.

When you are installing the Solaris software, you can choose to add or remove packages from the Solaris software group that you selected. When you are selecting which packages to add or remove, you need to know about software dependencies and how the Solaris software is packaged.

The following table lists the Solaris software groups and the recommended amount of disk space that you need to install each group.

Note – The following disk space recommendations include space for the following items.

- Swap space
- Patches
- Additional software packages

You might find that the following software groups require less disk space than the amount that is listed in the following table.

TABLE 5–1 Disk Space Recommendations for Software Groups

Software Group	Recommended Disk Space
Entire Solaris Software Group Plus OEM Support	2.9 Gbytes
Entire Solaris Software Group	2.7 Gbytes

TABLE 5–1 Disk Space Recommendations for Software Groups *(Continued)*

Software Group	Recommended Disk Space
Developer Solaris Software Group	2.4 Gbytes
End User Solaris Software Group	2.0 Gbytes
Core System Support Software Group	2.0 Gbytes

SPARC: How Swap Space Is Allocated

By default, the Solaris installation programs allocate swap space by placing swap so that it starts at the first available disk cylinder (typically cylinder 0 on SPARC systems). This placement provides maximum space for the root (/) file system during the default disk layout and enables the growth of the root (/) file system during an upgrade. If you use the installation program's automatic layout of disk slices and avoid manually changing the size of the swap slice, the Solaris installation program allocates a default swap area of 512 Mbytes.

If you think you might need to expand the swap area in the future, you can place the swap slice so that it starts at another disk cylinder by using one of the following methods.

- For the Solaris Web Start and Solaris `suninstall` programs, you can customize the disk layout in cylinder mode and manually assign the swap slice to the desired location.
- For the Solaris custom JumpStart installation program, you can assign the swap slice in the profile file. For more information on the Solaris Custom JumpStart profile file, see "Creating a Profile" on page 277.

For an overview of the swap space, see "Configuring Additional Swap Space (Tasks)" in *System Administration Guide: Devices and File Systems*.

Requirements When Using the Solaris Installation CD

When you are installing or upgrading by using the Solaris Installation CD, there are special requirements for SPARC slices and x86 `fdisk` partitions. When you are installing from a DVD or a net installation image, these requirements are not necessary.

TABLE 5–2 Solaris Installation CD Requirements

Platform	Requirements
Slice requirements for upgrading	When you use the Solaris Installation CD and the Solaris Web Start program to upgrade, you must have a slice on the disk that does not store files. The `swap` slice is preferred, but you can use any slice that is not located in any of the "upgradable" root slices that are listed in `/etc/vfstab`. The size of this slice must be at least 512 Mbytes.
x86 systems `fdisk` partition requirements	When you use the Solaris Installation CD, the Solaris Web Start program requires two `fdisk` partitions on the system disk to perform an installation or upgrade. ■ Solaris `fdisk` partition This is the typical Solaris `fdisk` partition. If you do not have a Solaris `fdisk` partition on your system, the Solaris Web Start program prompts you to create one. Caution: If you change the size of an existing `fdisk` partition, all data on that partition is automatically deleted. Back up your data before you create a Solaris `fdisk` partition. ■ x86 boot `fdisk` partition This is a 10–Mbyte `fdisk` partition that enables the x86 architecture to boot the miniroot that is placed on the newly created swap slice that is located on the Solaris `fdisk` partition. Caution: Do not create the x86 boot partition manually. The Solaris Web Start installation program creates the x86 boot partition by removing 10 Mbytes from the Solaris `fdisk` partition. By allowing the installation program to create the x86 boot partition, you prevent any existing `fdisk` partitions from being altered. **Note –** If you install or upgrade a system that has a Service partition, the Solaris Web Start installation program preserves the Service partition and creates the Solaris and x86 boot `fdisk` partitions. For more information on preserving a Service partition, see "x86: Change in Default Boot-Disk Partition Layout" on page 48.
x86 system upgrade limitations	When you use the Solaris Installation CD, you cannot use the Solaris Web Start program to upgrade from the Solaris 2.6 or Solaris 7 operating environments. The Solaris Installation CD requires a separate 10–Mbyte x86 boot partition that was not required in the Solaris 2.6 or Solaris 7 releases. You must use the Solaris Web Start program from a DVD or a net installation image, or use the Solaris `suninstall` program or custom JumpStart to upgrade.

TABLE 5–2 Solaris Installation CD Requirements *(Continued)*

Platform	Requirements

x86 systems logical block addressing requirement	Do not use the Solaris Installation CD unless your system can boot across the 1024–cylinder limit. Logical block addressing (LBA) enables the machine to boot beyond the 1024–cylinder limit and across Solaris disk slices. Use the Solaris Installation CD when your system's BIOS and SCSI driver for the default boot disk supports LBA.
	To determine if your system supports LBA, type:
	`#` **`prtconf -pv | grep -i lba`**
	`If the BIOS and SCSI driver for the default boot disk support LBA, the following message appears.`
	`lba-access-ok:`
	`If the SCSI driver for the default boot disk does not support LBA, the following message appears.`
	`no-bef-lba-access`
	`If the BIOS and SCSI driver for the default boot disk do not support LBA, use the Solaris DVD or a net installation image to install or upgrade.`

x86: Change in Default Boot-Disk Partition Layout

In the Solaris 9 (*x86 Platform Edition*) operating environment, the Solaris Web Start and `suninstall` installation programs use a new default boot-disk partition layout to accommodate the Service partition. If your system currently includes a Service partition, the new default boot-disk partition layout enables you to preserve this partition.

Note – If you install the Solaris 9 (*x86 Platform Edition*) operating environment on a system that does not currently include a Service partition, the installation program does not create a new Service partition by default. If you want to create a Service partition on your system, see your hardware documentation.

The new default includes the following partitions.

- First partition – Service partition (existing size on system)
- Second partition – x86 boot partition (approximately 11 Mbytes)

- Third partition – Solaris operating environment partition (remaining space on the boot disk)

If you want to use this default layout, select Default when the installation program asks you to choose a boot-disk layout.

CHAPTER 6

Gathering Information Before Installation or Upgrade (Planning)

This chapter contains checklists and worksheets to help you gather all of the information that you need to install or upgrade your system.

Checklist for Installation

Use the following checklist to prepare to install the Solaris operating environment.

- If you intend to install the Solaris software on a system through a `tip`(1) line, ensure that your window display is at least 80 columns wide and 24 rows long.

  To determine the current dimensions of your `tip` window, use the `stty` command. For more information, see the man page, `stty`(1).
- If the system is part of a network, verify that an Ethernet connector or similar network adapter is plugged into your system.
- If you are installing with the Solaris Web Start program by using the Solaris Installation CD, review "Requirements When Using the Solaris Installation CD" on page 46.
- Verify that you have enough disk space. For more information, refer to Chapter 5.
- SPARC: When you are using DVD media and are asked to boot from the `ok` prompt, always type the following command: **`boot cdrom`**.
- Review the *Solaris 9 9/04 Release Notes* and vendor release notes to ensure that the software you use is supported in the new Solaris release.
- Review the *Solaris 9 Sun Hardware Platform Guide* to ensure that your hardware is supported.

- Review the documentation that came with your system to ensure that your system and devices are supported by the Solaris release.

Worksheet for Installation

Use the following worksheet to gather the information that you need to install the Solaris operating environment. You do not need to gather all of the information that is requested on the worksheet. You only need to collect the information that applies to your system.

Note – Ignore the shaded rows if you are installing a standalone, non-networked system.

TABLE 6–1 Installation Worksheet

Information for Install	Description or Example	Answer
Network	Is the system connected to a network?	Yes/No
DHCP	Can the system use Dynamic Host Configuration Protocol (DHCP) to configure its network interfaces?	Yes/No
Host Name	Host name that you choose for the system.	
IP Address	If you are not using DHCP, supply the IP address for the system. Example: 192.168.9.1	
Subnet	If you are not using DHCP, is the system part of a subnet? If yes, what is the netmask of the subnet? Example: 255.255.0.0	Yes/No
IPv6	Do you want to enable IPv6 on this machine?	Yes/No

TABLE 6–1 Installation Worksheet *(Continued)*

Information for Install	Description or Example	Answer
Kerberos	Do you want to configure Kerberos security on this machine? If yes, gather this information: Default Realm: Administration Server: First KDC: (Optional) Additional KDCs:	Yes/No
Name Service	Which name service should this system use?	NIS+/NIS/DNS/LDAP/None
Domain Name	If the system uses a name service, supply the name of the domain in which the system resides.	
NIS+ and NIS	Do you want to specify a name server or let the installation program find one? If you want to specify a name server, provide the following information. Server's host name: Server's IP Address:	Specify One/Find One
DNS	Provide IP addresses for the DNS server. You must enter at least one IP address, but you can enter up to three addresses. Server's IP Address(es): You can enter a list of domains to search when a DNS query is made. Search Domain: Search Domain: Search Domain:	

TABLE 6–1 Installation Worksheet *(Continued)*

Information for Install	Description or Example	Answer
LDAP	Provide the following information about your LDAP profile. Profile Name: Profile Server: If you specify a proxy credential level in your LDAP profile, gather this information. Proxy-bind distinguished name: Proxy-bind password:	
Default Route	Do you want to specify a default route IP address or let the Solaris Web Start installation program or Solaris `suninstall` program find one? The default route provides a bridge that forwards traffic between two physical networks. An IP address is a unique number that identifies each host on a network. You have the following choices: ■ You can specify the IP address. An `/etc/defaultrouter` file is created with the specified IP address. When the system is rebooted, the specified IP address becomes the default route. ■ You can let the software detect an IP address. ■ The Solaris `suninstall` program detects an IP address when the system is rebooted. ■ The Solaris Web Start installation program can detect a default route. However, the system must be on a subnet that has a router that advertises itself by using the ICMP router discovery protocol. ■ You can choose None if you do not have a router or do not want the software to detect an IP address at this time. The software automatically tries to detect an IP address on reboot. Host IP address:	Specify One/Detect One/None
Time Zone	How do you want to specify your default time zone?	Geographic region Offset from GMT Time zone file

TABLE 6–1 Installation Worksheet *(Continued)*

Information for Install	Description or Example	Answer
Locales	For which geographic regions do you want to install support?	
SPARC: Power Management (only available on SPARC systems that support Power Management)	Do you want to use Power Management? **Note –** If your system has Energy Star version 3 or later, you are not prompted for this information.	Yes/No
Proxy Server Configuration (only available in the Solaris Web Start program)	Do you have a direct connection to the Internet or do you need to use a proxy server to gain access to the Internet? If you use a proxy server, provide the following information. Host: Port:	Direct Connection/Proxy Server
Automatic reboot or CD/DVD ejection	Reboot automatically after software installation? Eject CD/DVD automatically after software installation?	Yes/No Yes/No
Software Group	Which Solaris Software Group do you want to install?	Entire Plus OEM Entire Developer End User Core
Custom Package Selection	Do you want to add or remove software packages from the Solaris Software Group that you install? **Note –** When you select which packages to add or remove, you need to know about software dependencies and how Solaris software is packaged.	
SPARC: 64–bit	Do you want to install support for 64–bit applications?	Yes/No
Select Disks	On which disks do you want to install the Solaris software? Example: `c0t0d0`	

TABLE 6–1 Installation Worksheet *(Continued)*

Information for Install	Description or Example	Answer
x86: `fdisk` partitioning	Do you want to create, delete, or modify a Solaris fdisk partition? Each disk that is selected for file system layout must have a Solaris `fdisk` partition. Only one x86 Boot partition is allowed per system. If your system currently has a Service partition, the Solaris Web Start and `suninstall` installation programs preserve the Service partition by default. If you do not want to preserve the Service partition, you must customize the `fdisk` partitions. For more information on preserving a Service partition, see "x86: Change in Default Boot-Disk Partition Layout" on page 48.	
	Select Disks for `fdisk` Partition Customization?	Yes/No
	Customize `fdisk` partitions?	Yes/No
Preserve Data	Do you want to preserve any data that exists on the disks where you are installing the Solaris software?	Yes/No
Auto-layout File Systems	Do you want the installation program to automatically lay out file systems on your disks? If yes, which file systems should be used for auto-layout? Example: `/, /opt, /var` If no, you must provide file system configuration information.	Yes/No
Mount Remote File Systems (only available in the Solaris `suninstall` program)	Does this system need to access software on another file system? If yes, provide the following information about the remote file system.	Yes/No
	Server:	
	IP Address:	
	Remote File System:	
	Local Mount Point:	

Checklist for Upgrading

Use the following checklist to prepare to upgrade the Solaris operating environment.

- If you intend to upgrade the Solaris software on a system through a `tip`(1) line, ensure that your window display is at least 80 columns wide and 24 rows long.

  To determine the current dimensions of your `tip` window, use the `stty` command. For more information, see the man page, `stty`(1).
- If the system is part of a network, verify that an Ethernet connector or similar network adapter is connected to your system.
- If you are using the Solaris Web Start program from the Solaris Installation CD, verify that you have a 512–Mbyte slice on the disk. For detailed information, including more requirements for x86 systems, refer to Table 5–2. You can ignore these requirements if you are installing from a DVD or an installation image.
- If you are using Solaris Live Upgrade, determine your resource requirements for creating a new boot environment and upgrading it. For detailed information, refer to Chapter 34.
- If you are using Solaris Live Upgrade with Solaris Volume Manager, turn root mirroring off. For detailed information, refer to "Guidelines for Selecting Slices for File Systems" on page 401.
- Verify that you have enough disk space. For more information, refer to Chapter 5.
- SPARC: When you are using DVD media and are asked to boot from the `ok` prompt, always type the following command: **`boot cdrom`**.
- Review the *Solaris 9 9/04 Release Notes* and vendor release notes to ensure that the software you use is still supported in the new release.
- Review the *Solaris 9 Sun Hardware Platform Guide* to ensure that your hardware is supported.
- Review the documentation that came with your system to make sure your system and devices are supported by the Solaris release.
- Review vendor and third-party software documentation for additional upgrade instructions.
- Check for all of the available patches that you might need. The most recent patch list is provided at http://sunsolve.sun.com.
- Check the system for the existence of Prestoserve software. If you begin the upgrade process by shutting down the system with the `init 0` command, you might lose data. Refer to the Prestoserve documentation for shutdown instructions.
- x86: If you are using the Linux operating system, the Solaris `fdisk` partition and the Linux `swap` partition use the same identifier, 0x82. To resolve the problem, you can do one of the following.

- Choose not to use a Linux swap partition at all, provided that you have enough memory.
- Put the Linux swap partition on another drive.
- Back up the Linux data you want to keep to storage media, install the Solaris operating environment, and *then* reinstall Linux.

Caution – If you decide to install Linux after the Solaris operating environment, when the Linux installation program asks if you want to format the Linux swap partition (actually the Solaris fdisk partition) as a swap file, reply no.

Worksheet for Upgrading

Use the following worksheet to gather the information that you need to upgrade the Solaris operating environment. You do not need to gather all of the information that is requested on the worksheet. You only need to collect the information that applies to your system. If you are performing the upgrade over the network, the installation program provides the information for you, based on the current system configuration.

You cannot change basic system identification, such as host name or IP address. The installation program might prompt you for basic system identification, but you must enter the original values. If you use the Solaris Web Start program to upgrade, the upgrade fails if you attempt to change any of the values.

Note – Ignore the shaded rows if you are upgrading a standalone, nonnetworked system.

TABLE 6–2 Upgrade Worksheet

Information for Install	Description or Example	Answer
Network	Is the system connected to a network?	Yes/No
DHCP	Can the system use Dynamic Host Configuration Protocol (DHCP) to configure its network interfaces?	Yes/No
Host Name	Host name that you choose for the system.	

TABLE 6–2 Upgrade Worksheet *(Continued)*

Information for Install	Description or Example	Answer
IP Address	If you are not using DHCP, supply the IP address for the system. Example: 192.168.9.1	
Subnet	If you are not using DHCP, is the system part of a subnet? If yes, what is the netmask of the subnet? Example: 255.255.0.0	Yes/No
IPv6	Do you want to enable IPv6 on this machine?	Yes/No
Kerberos	Do you want to configure Kerberos security on this machine? If yes, gather this information: Default Realm: Administration Server: First KDC: (Optional) Additional KDCs:	Yes/No
Name Service	Which name service should this system use?	NIS+/NIS/DNS/LDAP/None
Domain Name	If the system uses a name service, supply the name of the domain in which the system resides.	
NIS+ and NIS	Do you want to specify a name server or let the installation program find one? If you want to specify a name server, provide the following information. Server's host name: Server's IP Address:	Specify One/Find One
DNS	Provide IP addresses for the DNS server. You must enter at least one IP address, but you can enter up to three addresses. Server's IP Address(es): You can enter a list of domains to search when a DNS query is made. Search Domain:	

TABLE 6–2 Upgrade Worksheet *(Continued)*

Information for Install	Description or Example	Answer
LDAP	Provide the following information about your LDAP profile. Profile Name: Profile Server: IP Address: If you specify a proxy credential level in your LDAP profile, gather this information. Proxy-bind distinguished name: Proxy-bind password:	
Default Route	Do you want to specify a default route IP address or let the Solaris Web Start installation program or Solaris `suninstall` program find one? The default route provides a bridge that forwards traffic between two physical networks. An IP address is a unique number that identifies each host on a network. You have the following choices: ■ You can specify the IP address. An `/etc/defaultrouter` file is created with the specified IP address. When the system is rebooted, the specified IP address becomes the default route. ■ You can let the software detect an IP address. ■ The Solaris `suninstall` program detects an IP address when the system is rebooted. ■ The Solaris Web Start installation program can detect a default route. However, the system must be on a subnet that has a router that advertises itself by using the ICMP router discovery protocol. ■ You can choose None if you do not have a router or do not want the software to detect an IP address at this time. The software automatically tries to detect an IP address on reboot. Host IP address:	Specify One/Find One/None
Time Zone	How do you want to specify your default time zone?	Geographic region Offset from GMT Time zone file

TABLE 6–2 Upgrade Worksheet *(Continued)*

Information for Install	Description or Example	Answer
Locales	For which geographic regions do you want to install support?	
SPARC: Power Management (only available on SPARC systems that support Power Management)	Do you want to use Power Management? **Note –** If your system has Energy Star version 3 or later, you are not prompted for this information.	Yes/No
Proxy Server Configuration (only available in the Solaris Web Start program)	Do you have a direct connection to the Internet or do you need to use a proxy server to gain access to the Internet? If you use a proxy server, provide the following information. Host: Port:	Direct Connection/Proxy Server
Automatic reboot or CD/DVD ejection	Reboot automatically after software installation? Eject CD/DVD automatically after software installation?	Yes/No Yes/No
Disk Space Reallocation	Do you want the installation program to automatically re-lay out file systems on your disks? If yes, which file systems should be used for auto-layout? Example: `/`, `/opt`, `/var` If no, you must provide information for file system configuration.	Yes/No
SPARC: 64–bit	Do you want to install support for 64–bit applications?	Yes/No

CHAPTER 7

Preconfiguring System Configuration Information (Tasks)

This chapter describes how to preconfigure system information. Preconfiguration can help you to avoid being prompted for this information when you install the Solaris operating environment. This chapter also describes how to preconfigure Power Management™ information. This chapter contains the following sections:

- "Advantages of Preconfiguring System Configuration Information" on page 63
- "Ways to Preconfigure System Configuration Information" on page 64
- "Preconfiguring With the `sysidcfg` File" on page 65
- "Preconfiguring With the Name Service" on page 82
- "Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85
- "SPARC: Preconfiguring Power Management Information" on page 95

Advantages of Preconfiguring System Configuration Information

The installation methods require configuration information about a system, such as peripheral devices, host name, Internet Protocol (IP) address, and name service. Before the installation tools prompt you for configuration information, they check for the information in the `sysidcfg` file and then in the name service databases.

When the Solaris Web Start program, Solaris `suninstall` program, or the custom JumpStart installation program detects preconfigured system information, the installation program does not prompt you to enter the information. For example, you have several systems and you do not want a time zone prompt every time you install the Solaris 9 software on one of the systems. You can specify the time zone in the `sysidcfg` file or the name service databases. When you install the Solaris 9 software, the installation program does not prompt you to type a time zone.

Ways to Preconfigure System Configuration Information

You can choose one of the following ways to preconfigure system configuration information. You can add the system configuration information to either of the following.

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

If your site uses DHCP, you can also preconfigure some system information in the site DHCP server. For more information about how to use a DHCP server to preconfigure system information, see “Preconfiguring System Configuration Information With the DHCP Service (Tasks)” on page 85.

Use the following table to determine whether to use a `sysidcfg` file or a name service database to preconfigure system configuration information.

TABLE 7–1 Methods to Preconfigure System Configuration Information

Preconfigurable System Information	Preconfigurable With the `sysidcfg` File?	Preconfigurable With the Name Service?
Name service	Yes	Yes
Domain name	Yes	No
Name server	Yes	No
Network interface	Yes	No
Host name	Yes Because this information is system specific, edit the name service rather than create a different `sysidcfg` file for each system.	Yes
Internet Protocol (IP) address	Yes Because this information is system specific, edit the name service rather than create a different `sysidcfg` file for each system.	Yes
Netmask	Yes	No
DHCP	Yes	No

TABLE 7–1 Methods to Preconfigure System Configuration Information *(Continued)*

Preconfigurable System Information	Preconfigurable With the `sysidcfg` File?	Preconfigurable With the Name Service?
IPv6	Yes	No
Default route	Yes	No
Root password	Yes	No
Security policy	Yes	No
Language (locale) in which to display the install program and desktop	Yes	Yes, if NIS or NIS+ No, if DNS or LDAP
Terminal type	Yes	No
Time zone	Yes	Yes
Date and time	Yes	Yes
Web proxy	No You can configure this information during a Solaris Web Start installation, but not through the `sysidcfg` file or the name service.	No
x86: Monitor type	Yes	No
x86: Keyboard language, keyboard layout	Yes	No
x86: Graphics card, color depth, display resolution, screen size	Yes	No
x86: Pointing device, number of buttons, IRQ level	Yes	No
SPARC: Power Management (autoshutdown) You cannot preconfigure Power Management through the `sysidcfg` file or the name service. "SPARC: Preconfiguring Power Management Information" on page 95 contains details.	No	No

Preconfiguring With the `sysidcfg` File

You can specify a set of keywords in the `sysidcfg` file to preconfigure a system. The keywords are described in "`sysidcfg` File Keywords" on page 67.

You must create a unique sysidcfg file for every system that requires different configuration information. You can use the same sysidcfg file to preconfigure the time zone on a set of systems if you want all the systems to be assigned the same time zone. However, if you want to preconfigure a different root (superuser) password for each of those systems, you need to create a unique sysidcfg file for each system.

You can place the sysidcfg file in one of the following.

- NFS file system – If you put the sysidcfg file in a shared NFS file system, you must use the -p option of the add_install_client(1M) command when you set up the system to install from the network. The -p option specifies where the system can find the sysidcfg file when you install the Solaris 9 software.
- UFS or PCFS diskette – Place the sysidcfg file in the root (/) directory on the diskette.
- HTTP or HTTPS server – If you want to perform a WAN boot installation, place the sysidcfg file in the document root directory of the web server.

Note – If you are performing a custom JumpStart installation and you want to use a sysidcfg file on a diskette, you must place the sysidcfg file on the profile diskette. To create a profile diskette, see “Creating a Profile Diskette for Standalone Systems” on page 269.

You can place only one sysidcfg file in a directory or on a diskette. If you are creating more than one sysidcfg file, you must place each file in a different directory or on a different diskette.

Syntax Rules for the sysidcfg File

You can use two types of keywords in the sysidcfg file: independent and dependent. Dependent keywords are guaranteed to be unique only within independent keywords. A dependent keyword exists only when it is identified with its associated independent keyword.

In this example, name_service is the independent keyword, while domain_name and name_server are the dependent keywords:

```
name_service=NIS {domain_name=marquee.central.example.com
name_server=connor(192.168.112.3)}
```

Syntax Rule	Example

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

Keywords are not case sensitive.	`TIMEZONE=US/Central`
	`terminal=sun-cmd`

Enclose all dependent keywords in curly braces ({}) to tie them to their associated independent keyword.	`name_service=NIS`
	`   {domain_name=marquee.central.example.com`
	`    name_server=connor(192.168.112.3)}`

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

For all keywords except the `network_interface` keyword, only one instance of a keyword is valid. However, if you specify the keyword more than once, only the first instance of the keyword is used.	`name_service=NIS`
	`name_service=DNS`

`sysidcfg` File Keywords

Table 7–2 lists the keywords you can use to configure system information in the `sysidcfg` file.

TABLE 7–2 Keywords You Can Use in `sysidcfg`

Configuration Information	Keyword
Name service, domain name, name server	“`name_service` Keyword” on page 68
Network interface, host name, Internet Protocol (IP) address, netmask, DHCP, IPv6	“`network_interface` Keyword” on page 71
Root password	“`root_password` Keyword” on page 76
Security policy	“`security_policy` Keyword” on page 76
Language in which to display the install program and desktop	“`system_locale` Keyword” on page 77
Terminal type	“`terminal` Keyword” on page 77
Time zone	“`timezone` Keyword” on page 77
Date and time	“`timeserver` Keyword” on page 78

TABLE 7–2 Keywords You Can Use in sysidcfg *(Continued)*

Configuration Information	Keyword
x86: Monitor type	"x86: monitor Keyword" on page 78
x86: Keyboard language, keyboard layout	"x86: keyboard Keyword" on page 78
x86: Graphics card, screen size, color depth, display resolution	"x86: display Keyword" on page 79
x86: Pointing device, number of buttons, IRQ level	"x86: pointer Keyword" on page 79

The following sections describe the keywords you can use in the sysidcfg file.

name_service Keyword

You can use the name_service keyword to configure the name service, the domain name, and the name server for the system. The following sample shows the general syntax for the name_service keyword.

```
name_service=name-service {domain_name=domain-name
                                  name_server=name-server
                                  optional-keyword=value}
```

Choose only one value for name_service. Include all or none of the domain_name,name_server, or optional keywords, as needed. If no keywords are used, omit the curly braces {}.

The following sections describe the keyword syntax to configure the system to use a specific name service.

NIS Syntax for name_service Keyword

Use the following syntax to configure the system to use the NIS name service.

```
name_service=NIS {domain_name=domain-name
                   name_server=hostname(ip-address)}
```

domain-name	Specifies the domain name
hostname	Specifies the host name of the name server
ip-address	Specifies the IP address of the name server

EXAMPLE 7–1 Specifying a NIS Server With the name_service Keyword

The following example specifies a NIS server with the domain name west.example.com. The server's host name is timber, and the server IP address is 192.168.2.1.

```
name_service=NIS {domain_name=west.example.com
                  name_server=timber(192.168.2.1)}
```

For more information about the NIS name service, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

NIS+ Syntax for `name_service` Keyword

Use the following syntax to configure the system to use the NIS name service.

```
name_service=NIS+ {domain_name=domain-name
                   name_server=hostname(ip-address)}
```

domain-name Specifies the domain name

hostname Specifies the host name of the name server

ip-address Specifies the IP address of the name server

EXAMPLE 7–2 Specifying a NIS+ Server With the `name_service` Keyword

The following example specifies a NIS+ server with the domain name `west.example.com`. The server's host name is `timber`, and the server IP address is 192.168.2.1.

```
name_service=NIS+ {domain_name=west.example.com
                   name_server=timber(192.168.2.1)}
```

For more information about the NIS+ name service, see *System Administration Guide: Naming and Directory Services (FNS and NIS+)*.

DNS Syntax for `name_service` Keyword

Use the following syntax to configure the system to use DNS.

```
name_service=DNS {domain_name=domain-name
                  name_server=ip-address,ip-address,ip-address
                  search=domain-name,domain-name,domain-name,
                  domain-name,domain-name,domain-name}
```

domain_name=*domain-name* Specifies the domain name.

name_server=*ip-address* Specifies the IP address of the DNS server. You can specify up to three IP addresses as values for the `name_server` keyword.

search=*domain-name* (Optional) Specifies additional domains to search for name service information. You can specify up to six domain names to search. The total length of each search entry cannot exceed 250 characters.

EXAMPLE 7–3 Specifying a DNS Server With the `name_service` Keyword

The following example specifies a DNS server with the domain name `west.example.com`. The server IP addresses are 10.0.1.10 and 10.0.1.20. `example.com` and `east.example.com` are listed as additional domains to search for name service information.

```
name_service=DNS {domain_name=west.example.com
                 name_server=10.0.1.10,10.0.1.20
                 search=example.com,east.example.com}
```

For more information about the DNS name service, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

LDAP Syntax for `name_service` Keyword

Use the following syntax to configure the system to use LDAP.

```
name_service=LDAP {domain_name=domain_name
                  profile=profile_name profile_server=ip_address
                  proxy_dn="proxy_bind_dn" proxy_password=password}
```

domain_name Specifies the domain name of the LDAP server.

profile_name Specifies the name of the LDAP profile you want to use to configure the system.

ip_address Specifies the IP address of the LDAP profile server.

proxy_bind_dn (Optional) Specifies the proxy bind distinguished name. You must enclose the *proxy_bind_dn* value in double quotes.

password (Optional) Specifies the client proxy password.

EXAMPLE 7–4 Specifying an LDAP Server With the `name_service` Keyword

The following example specifies an LDAP server with the following configuration information.

- The domain name is `west.example.com`.
- The installation program uses the LDAP profile that is named `default` to configure the system.
- The IP address of the LDAP server is 172.221.2.1.
- The proxy bind distinguished name includes the following information.
  - The common name for the entry is `proxyagent`.
  - The organizational unit is `profile`.
  - The proxy domain includes the `west`, `example`, and `com` domain components.
- The poxy password is `password`.

```
name_service=LDAP {domain_name=west.example.com
                   profile=default
                   profile_server=172.31.2.1
                   proxy_dn="cn=proxyagent,ou=profile,
                   dc=west,dc=example,dc=com"
                   proxy_password=password}
```

For more information about how to use LDAP, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

`network_interface` Keyword

Use the `network_interface` keyword to perform the following tasks.

- Specify a host name
- Specify an IP address
- Specify a netmask value
- Use DHCP to configure the network interface
- Enable IPv6 on the network interface

The following sections describe how to use the `network_interface` keyword to configure the system interfaces.

Syntax for Non-Networked Systems

To turn off networking for the system, set the `network_interface` value to none. For example:

```
network_interface=none
```

Syntax for Configuring a Single Interface

You can use the `network_interface` keyword to configure a single interface in the following ways.

- **With DHCP** – You can use a DHCP server on your network to configure the network interface. For more information on how to use a DHCP server during your installation, see "Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85.

  To use the DHCP server to configure a single interface on the system, use the following syntax for the `network_interface` keyword.

  ```
  network_interface=PRIMARY or value
                        {dhcp protocol_ipv6=yes-or-no}
  ```

`PRIMARY`
Instructs the installation program to configure the first up, non-loopback interface that is found on the system. The order is the same as the order that is displayed with the `ifconfig` command. If no interfaces are up, then the first non-loopback interface is used. If no non-loopback interfaces are found, then the system is non-networked.

value
Instructs the installation program to configure a specific interface, such as `hme0` or `eri1`.

`protocol_ipv6=`*yes-or-no*
Instructs the installation program to configure the system to either use IPv6 or to not use IPv6.

For WAN boot installations, you must set the value of `protocol_ipv6=no`.

- **Without DHCP** – If you do not want to use DHCP to configure the network interface, you can specify the configuration information in the `sysidcfg` file. To instruct the installation program to configure a single interface on the system without using DHCP, use the following syntax.

```
network_interface=PRIMARY or value
                   {hostname=host_name
                    default_route=ip_address
                    ip_address=ip_address
                    netmask=netmask
                    protocol_ipv6=yes_or_no}
```

PRIMARY
Instructs the installation program to configure the first up, non-loopback interface that is found on the system. The order is the same as the order that is displayed with the `ifconfig` command. If no interfaces are up, then the first non-loopback interface is used. If no non-loopback interfaces are found, then the system is not networked.

Note – Do not use the `PRIMARY` keyword value if you want to configure multiple interfaces.

value
Instructs the installation program to configure a specific interface, such as `hme0` or `eri1`.

`hostname=`*host_name*
(Optional) Specifies the host name of the system.

`default_route=`*ip_address*
(Optional) Specifies the IP address of the default router. If you want the installation program to detect the router by using the ICMP router discovery protocol, omit this keyword.

Note – If the installation program cannot detect the router, you are prompted for the router information during the installation.

`ip_address=`*ip_address* (Optional) Specifies the IP address of the system.

`netmask=`*netmask* (Optional) Specifies the netmask value for the system.

`protocol_ipv6=`*yes_or_no* (Optional) Instructs the installation program to configure the system to either use IPv6 or to not use IPv6.

Note – To perform an unattended custom JumpStart installation, you must specify a value for the `protocol_ipv6` keyword.

For WAN boot installations, you must set the value of `protocol_ipv6=no`.

Include any combination or none of the `hostname`, `ip_address`, and `netmask` keywords, as needed. If you do not use any of these keywords, omit the curly braces ({}).

EXAMPLE 7–5 Configuring a Single Interface By Using DHCP With the `network_interface` Keyword

The following example instructs the installation program to use DHCP to configure the `eri0` network interface. IPv6 support is not enabled.

```
network_interface=eri0 {dhcp protocol_ipv6=no}
```

EXAMPLE 7–6 Configuring a Single Interface By Specifying Configuration Information With the `network_interface` Keyword

The following example configures the interface `eri0` with the following settings.

- The host name is set to host1.
- The IP address is set to 172.31.88.100.
- The netmask is set to 255.255.255.0.
- IPv6 support is not enabled on the interface.

```
network_interface=eri0 {hostname=host1 ip_address=172.31.88.100
                        netmask=255.255.255.0 protocol_ipv6=no}
```

Syntax for Configuring Multiple Interfaces

You can configure multiple network interfaces in your `sysidcfg` file. For each interface that you want to configure, include a `network_interface` entry in the `sysidcfg` file.

You can use the `network_interface` keyword to configure multiple interfaces in the following ways.

- **With DHCP** – You can use a DHCP server on your network to configure a network interface. For more information on how to use a DHCP server during your installation, see "Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85.

  To use the DHCP server to configure a network interface on the system, use the following syntax for the `network_interface` keyword.

  ```
  network_interface=value {primary
                                dhcp protocol_ipv6=yes-or-no}
  ```

  value
  Instructs the installation program to configure a specific interface, such as `hme0` or `eri1`.

  `primary`
  (Optional) Specifies *value* as the primary interface.

  `protocol_ipv6=`*yes-or-no*
  Instructs the installation program to configure the system to either use IPv6 or to not use IPv6.

  Note – For WAN boot installations, you must set the value of `protocol_ipv6=no`.

- **Without DHCP** – If you do not want to use DHCP to configure the network interface, you can specify the configuration information in the `sysidcfg` file. To instruct the installation program to configure multiple interfaces without using DHCP, use the following syntax.

  ```
  network_interface=value {primary hostname=host_name
                                default_route=ip_address
                                ip_address=ip_address
                                netmask=netmask
                                protocol_ipv6=yes_or_no}
  ```

  value
  Instructs the installation program to configure a specific interface, such as `hme0` or `eri1`.

  `primary`
  (Optional) Specifies *value* as the primary interface.

  `hostname=`*host_name*
  (Optional) Specifies the host name of the system.

  `default_route=`*ip_address*
  (Optional) Specifies the IP address of the default router. If you want the installation program to

detect the router by using the ICMP router discovery protocol, omit this keyword.

Note – If the installation program cannot detect the router, you are prompted for the router information during the installation.

`ip_address=`*ip_address* (Optional) Specifies the IP address of the system.

`netmask=`*netmask* (Optional) Specifies the netmask value for the system.

`protocol_ipv6=`*yes_or_no* (Optional) Instructs the installation program to configure the system to either use IPv6 or to not use IPv6.

Note – To perform an unattended custom JumpStart installation, you must specify a value for the `protocol_ipv6` keyword.

For WAN boot installations, you must set the value of `protocol_ipv6=no`.

Include any combination or none of the `hostname`, `ip_address`, and `netmask` keywords, as needed. If you do not use any of these keywords, omit the curly braces ({}).

In the same `sysidcfg` file, you can use DHCP to configure certain interfaces, while also specifying the configuration information for other interfaces in the `sysidcfg` file.

EXAMPLE 7–7 Configuring Multiple Interfaces With the `network_interface` Keyword

In the following example, the network interfaces eri0 and eri1 are configured in the following way.

- `eri0` is configured by using the DHCP server. IPv6 support is not enabled on `eri0`.
- `eri1` is the primary network interface. The host name is set to host1, and the IP address is set to 172.31.88.100. The netmask is set to 255.255.255.0. IPv6 support is not enabled on `eri1`.

```
network_interface=eri0 {dhcp protocol_ipv6=no}
network_interface=eri1 {primary hostname=host1
                         ip_address=172.31.88.100
                         netmask=255.255.255.0
```

EXAMPLE 7–7 Configuring Multiple Interfaces With the `network_interface` Keyword *(Continued)*

```
                                protocol_ipv6=no}
```

`root_password` Keyword

You can specify the root password to the system in the `sysidcfg` file. To specify the root password, use the `root_password` keyword with the following syntax.

`root_password=`*encrypted-password*

encrypted-password is the encrypted password as it appears in the `/etc/shadow` file.

`security_policy` Keyword

You can use the `security_policy` keyword in your `sysidcfg` file to configure your system to use the Kerberos network authentication protocol. If you want to configure the system to use Kerberos, use the following syntax.

```
security_policy=kerberos {default_realm=FQDN
                                   admin_server=FQDN kdc=FQDN1, FQDN2, FQDN3}
```

FQDN specifies the fully qualified domain name of the Kerberos default realm, the administration server, or key distribution center (KDC). You must specify at least one, but no more than three, key distribution centers.

If you do not want to set the security policy for the system, set `security_policy=NONE`.

For more information about the Kerberos network authentication protocol, see *System Administration Guide: Security Services*.

EXAMPLE 7–8 Configuring the System to Use Kerberos With the `security_policy` Keyword

The following example configures the system to use Kerberos with the following information.

- The Kerberos default realm is `example.COM`.
- The Kerberos administration server is `krbadmin.example.COM`.
- The two key distribution centers are `kdc1.example.COM` and `kdc2.example.COM`.

```
security_policy=kerberos
                     {default_realm=example.COM
                      admin_server=krbadmin.example.COM
                      kdc=kdc1.example.COM,
                      kdc2.example.COM}
```

system_locale Keyword

You can use the system_locale keyword to specify the language in which to display the install program and desktop. Use the following syntax to specify a locale.

```
system_locale=locale
```

locale specifies the language you want the system to use to display the installation panels and screens. For a list of valid locale values, see the /usr/lib/locale directory or Appendix D.

terminal Keyword

You can use the terminal keyword to specify the terminal type for the system. Use the following syntax to specify the terminal type.

```
terminal=terminal_type
```

terminal_type specifies the terminal type for the system. For a list of valid terminal values, see the subdirectories in the /usr/share/lib/terminfo directory.

timezone Keyword

You can set the time zone for the system with the timezone keyword. Use the following syntax.

```
timezone=timezone
```

In the previous example, *timezone* specifies the time zone value for the sytem. The directories and files in the /usr/share/lib/zoneinfo directory provide the valid time zone values. The *timezone* value is the name of the path relative to the /usr/share/lib/zoneinfo directory. You can also specify any valid Olson time zone.

EXAMPLE 7–9 Configuring the System Time Zone Information With the timezone Keyword

In the following example, the system time zone is set to mountain standard time in the United States.

```
timezone=US/Mountain
```

The installation program configures the system to use the time zone information in /usr/share/lib/zoneinfo/US/Mountain.

`timeserver` Keyword

You can use the `timeserver` keyword to specify the system that sets the date and time on the system you want to install.

Note – Do not set `timeserver=`*hostname* `or` *ip-address* if you are running a name service.

Choose one of the following methods to set the `timeserver` keyword.

- To configure the system to serve as its own time server, set `timeserver=localhost`. If you specify `localhost` as the time server, the system's time is assumed to be correct.
- To specify another system as the time server, specify either the host name or the IP address of the time server with the `timeserver` keyword. Use the following syntax.

  ```
  timeserver=hostname or ip-address
  ```

  hostname is the host name of the time server system. *ip-address* specifies the IP address of the time server.

x86: `monitor` Keyword

For x86-based systems, you can configure the monitor information with the `monitor` keyword. Use the following syntax with the `monitor` keyword.

```
monitor=monitor_type
```

To set the value for the `monitor` keyword, run the `kdmconfig -d` command on the system you want to install. Copy the line of output that includes the `monitor` keyword, and include this line in the `sysidcfg` file.

For more information, see the `kdmconfig`(1M) man page.

x86: `keyboard` Keyword

For x86-based systems, you can configure the keyboard language and layout information with the `keyboard` keyword. Use the following syntax with the `keyboard` keyword.

```
keyboard=keyboard_language {layout=value}
```

To set the value for the keyboard keyword, run the kdmconfig -d command on the system you want to install. Copy the line of output that includes the keyboard keyword, and include this line in the sysidcfg file.

For more information, see the kdmconfig(1M) man page.

x86: display Keyword

For x86-based systems, you can configure the following information with the display keyword.

- Graphics card
- Screen size
- Color depth
- Display resolution

Use the following syntax with the display keyword.

```
display=graphics_card {size=screen_size
                            depth=color_depth
                            resolution=screen_resolution}
```

To set the appropriate values for the display keyword, run the kdmconfig -d command on the system you want to install. Copy the line of output that includes the display keyword, and include this line in the sysidcfg file.

For more information, see the kdmconfig(1M) man page.

x86: pointer Keyword

For x86-based systems, you can configure the following mouse information with the pointer keyword.

- Pointing device
- Number of buttons
- IRQ level

Use the following syntax with the pointer keyword.

```
pointer=pointing_device {nbuttons=number_buttons irq=value}
```

To set the value for the pointer keyword, run the kdmconfig -d command on the system you want to install. Copy the line of output that includes the pointer keyword, and include this line in the sysidcfg file.

For more information, see the kdmconfig(1M) man page.

▼ To Create a sysidcfg Configuration File

1. **Create a file called sysidcfg in a text editor.**
2. **Type the sysidcfg keywords you want.**
3. **Save the sysidcfg file.**

Note – If you create more than one sysidcfg file, you must save each file in a separate directory or on a separate diskette.

---

4. **Make the sysidcfg file available to clients through the following:**
   - A shared NFS file system. Use add_install_client(1M) with the -p option to set up the system to install from the network.
   - The root (/) directory on a UFS diskette or PCFS diskette.

EXAMPLE 7–10 SPARC: sysidcfg File

The following is an example of a sysidcfg file for a SPARC based system. The host name, IP address, and netmask of this system has been preconfigured by editing the name service. Because all of the system configuration information is preconfigured in this file, you can use a custom JumpStart profile to perform a custom JumpStart installation.

```
system_locale=en_US
timezone=US/Central
terminal=sun-cmd
timeserver=localhost
name_service=NIS {domain_name=marquee.central.example.com
                  name_server=connor(172.31.112.3)}
root_password=m4QPOWNY
network_interface=le0 {hostname=feron
                       default_route=192.168.88.1
                       ip_address=192.168.88.210
                       netmask=255.255.0.0
                       protocol_ipv6=no}
security_policy=kerberos {default_realm=example.COM
                          admin_server=krbadmin.example.COM
                          kdc=kdc1.example.COM,
                          kdc2.example.COM}
```

EXAMPLE 7–10 SPARC: sysidcfg File *(Continued)*

EXAMPLE 7–11 x86: sysidcfg File

The following sample sysidcfg file is for a group of x86 based systems that all use the same type of keyboard, graphics cards, and pointing devices. The device information (keyboard, display, and pointer) was obtained by running the kdmconfig(1M) command with the -d option. If the following example sysidcfg file is used, a prompt that asks you to select a language (system_locale) is displayed before installation can proceed.

```
keyboard=ATKBD {layout=US-English}
display=ati {size=15-inch}
pointer=MS-S
timezone=US/Central
timeserver=connor
terminal=ibm-pc
name_service=NIS {domain_name=marquee.central.example.com
                  name_server=connor(172.25.112.3)}
root_password=URFUni9
```

EXAMPLE 7–12 sysidcfg File for Configuring Multiple Interfaces

In the following sample sysidcfg file, configuration information is specified for both the eri0 and eri1 network interfaces. The eri0 interface is configured as the primary network interface, and eri1 is configured as a secondary network interface.

```
timezone=US/Pacific
system_locale=C
terminal=xterms
timeserver=localhost
network_interface=eri0 {primary
                        hostname=feron
                        ip_address=192.168.2.7
                        netmask=255.255.255.0
                        protocol_ipv6=no
                        default_route=192.168.2.1}

network_interface=eri1 {hostname=feron-b
                        ip_address=192.168.3.8
                        netmask=255.255.255.0
                        protocol_ipv6=no
                        default_route=192.168.3.1}
root_password=JE2C35JGZi4B2
security_policy=none
name_service=NIS {domain_name=domain.example.com
                  name_server=nis-server(192.168.2.200)}
```

Preconfiguring With the Name Service

The following table provides a high-level overview of the name service databases that you need to edit and populate to preconfigure system information.

System Information to Preconfigure	Name Service Database
Host name and Internet Protocol (IP) address	`hosts`
Date and time	`hosts`. Specify the `timehost` alias next to the host name of the system that will provide the date and time for the systems that are being installed.
Time zone	`timezone`
Netmask	`netmasks`

You cannot preconfigure the locale for a system with the DNS or LDAP name service. If you use the NIS or NIS+ name service, follow the procedure for your name service to preconfigure the locale for a system:

- “To Preconfigure the Locale Using NIS” on page 82
- “To Preconfigure the Locale Using NIS+” on page 84

▼ To Preconfigure the Locale Using NIS

1. **Become superuser on the name server.**

2. **Change `/var/yp/Makefile` to add the locale map.**

   a. **Insert this shell procedure after the last *variable*`.time` shell procedure.**

```
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
```

b. **Find the string `all:` and, at the end of the list of variables, insert the word `locale`.**

```
all: passwd group hosts ethers networks rpc services protocols \
    netgroup bootparams aliases publickey netid netmasks c2secure \
    timezone auto.master auto.home locale
```

c. **Near the end of the file, after the last entry of its type, insert the string `locale: locale.time` on a new line.**

```
passwd: passwd.time
group: group.time
hosts: hosts.time
ethers: ethers.time
networks: networks.time
rpc: rpc.time
services: services.time
protocols: protocols.time
netgroup: netgroup.time
bootparams: bootparams.time
aliases: aliases.time
publickey: publickey.time
netid: netid.time
passwd.adjunct: passwd.adjunct.time
group.adjunct: group.adjunct.time
netmasks: netmasks.time
timezone: timezone.time
auto.master: auto.master.time
auto.home: auto.home.time
locale: locale.time
```

d. **Save the file.**

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

locale domain_name

Or

locale system_name

Note – Appendix D contains a list of valid locales.

For example, the following entry specifies that French is the default language that is used in the `worknet.com` domain:

```
fr worknet.com
```

And the following entry specifies that Belgian French is the default locale that is used by a system named `charlie`:

```
fr_BE charlie
```

Note – Locales are available on the Solaris DVD or Solaris Software 1 of 2 CD.

---

4. **Make the maps:**

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

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

▼ To Preconfigure the Locale Using NIS+

The following procedure assumes the NIS+ domain is set up. Setting up the NIS+ domain is documented in the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP).*

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

2. **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 needed entries to the `locale`.**

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

   name Either the domain name or a specific system name for which you want to preconfigure a default locale.

   locale The locale you want to install on the system and use on the desktop after the system is rebooted. Appendix D contains a list of valid locales.

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

Note – Locales are available on the Solaris DVD or Solaris Software 1 of 2 CD.

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

Preconfiguring System Configuration Information With the DHCP Service (Tasks)

The Dynamic Host Configuration Protocol (DHCP) enables host systems in a TCP/IP network to be configured automatically for the network as they boot. DHCP uses a client and server mechanism. Servers store and manage configuration information for clients, and provide that information on a client's request. The information includes the client's IP address and information about network services available to the client.

A primary benefit of DHCP is its ability to manage IP address assignments through leasing. Leasing allows IP addresses to be reclaimed when not in use and reassigned to other clients. This ability enables a site to use a smaller pool of IP address than would be needed if all clients were assigned a permanent address.

You can use DHCP to install the Solaris operating environment on certain client systems on your network. Only Sun Enterprise Ultra systems and x86 systems that meet the hardware requirements for running the Solaris operating environment can use this feature.

The following task map shows the high-level tasks that must be performed to enable clients to obtain installation parameters by using DHCP.

TABLE 7–3 Task Map: Preconfiguring System Configuration Information With the DHCP Service

Task	Description	Instructions
Set up an install server.	Set up a Solaris server to support clients that must install the Solaris operating environment from the network.	Chapter 13

TABLE 7–3 Task Map: Preconfiguring System Configuration Information With the DHCP Service *(Continued)*

Task	Description	Instructions
Set up client systems for Solaris installation over the network by using DHCP.	Use `add_install_client -d` to add DHCP network installation support for a class of client (of a certain machine type, for example) or a particular client ID.	Using Solaris DVD: "Adding Systems to Be Installed From the Network With a DVD Image" on page 137 Using Solaris CD: "Adding Systems to Be Installed From the Network With a CD Image" on page 167 `add_install_client`(1M)
Prepare your network to use the DHCP service.	Decide how you want to configure your DHCP server.	"Planning for DHCP Service (Task)" in *System Administration Guide: IP Services*
Configure the DHCP server.	Use DHCP Manager to configure your DHCP server	"Configuring DHCP Service (Task)" in *System Administration Guide: IP Services*
Create DHCP options for installation parameters and macros that include the options.	Use DHCP Manager or `dhtadm` to create new Vendor options and macros that the DHCP server can use to pass installation information to the clients.	"Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86

Creating DHCP Options and Macros for Solaris Installation Parameters

When you add clients with the `add_install_client -d` script on the install server, the script reports DHCP configuration information to standard output. This information can be used when you create the options and macros that are needed to pass network installation information to clients.

To install DHCP clients from the network, you must create Vendor category options to pass information that is needed to install the Solaris operating environment. Table 7–4 shows the options you must create and the properties that are needed to create them.

You can customize the macros in your DHCP service to perform the following types of installations.

- **Class-specific installations** - You can instruct the DHCP service to perform a network installation for all clients of a specific class. For example, you can define a DHCP macro that performs the same installation on all Sun Blade systems on the network. Use the output of the `add_install_client -d` command to set up a class-specific installation.
- **Client-specific installations** - You can instruct the DHCP service to perform a network installation for a client with a specific ethernet address. For example, you can define a DHCP macro that performs a specific installation on the client with the ethernet address 00:07:e9:04:4a:bf. Use the output of the `add_install_client -d -e` *ethernet_address* command to set up a client-specific installation.

For more information on setting up clients to install from the network, see the following procedures.

- For network installations that use DVD media, see "Adding Systems to Be Installed From the Network With a DVD Image" on page 137.
- For network installations that use CD media, see "Adding Systems to Be Installed From the Network With a CD Image" on page 167.

The vendor client classes that are listed in the following table determine what classes of client can use the option. Vendor client classes that are listed here are examples only. You should specify client classes that indicate the actual clients in your network that you need to install from the network. See "Working With DHCP Options (Task Map)" in *System Administration Guide: IP Services* for information about how to determine a client's vendor client class.

For detailed information on DHCP options, see "DHCP Option Information" in *System Administration Guide: IP Services*.

TABLE 7–4 Values for Creating Vendor Category Options for Solaris Clients

Name	Code	Data Type	Granularity	Maximum	Vendor Client Classes *	Description
The following Vendor category options are required to enable a DHCP server to support Solaris installation clients. The options are used in the Solaris client's startup scripts.						

`SrootIP4`	2	IP address	1	1	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						IP address of root server

`SrootNM`	3	ASCII text	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						Host name of root server

`SrootPTH`	4	ASCII text	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						Path to the client's root directory on the root server

TABLE 7–4 Values for Creating Vendor Category Options for Solaris Clients *(Continued)*

Name	Code	Data Type	Granularity	Maximum	Vendor Client Classes *	Description
`SinstIP4`	10	IP address	1	1	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	IP address of JumpStart install server
`SinstNM`	11	ASCII text	1	0	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	Host name of install server
`SinstPTH`	12	ASCII text	1	0	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	Path to installation image on install server
The following options can be used by the client startup scripts, but are not required by the scripts.						
`SrootOpt`	1	ASCII text	1	0	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	NFS mount options for the client's root file system
`SbootFIL`	7	ASCII text	1	0	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	Path to the client's boot file
`SbootRS`	9	NUMBER	2	1	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	NFS read size used by standalone boot program when loading the kernel
`SsysidCF`	13	ASCII text	1	0	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	Path to `sysidcfg` file, in the format *server:/path*
`SjumpsCF`	14	ASCII text	1	0	`SUNW.Sun-Blade-1000, SUNW.Sun-Fire-880, SUNW.i86pc`	Path to JumpStart configuration file in the format *server:/path*

TABLE 7–4 Values for Creating Vendor Category Options for Solaris Clients *(Continued)*

Name	Code	Data Type	Granularity	Maximum	Vendor Client Classes *	Description

`SbootURI`	16	ASCII text	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						Path to the standalone boot file or path to the WAN boot file. For the standalone boot file, use the following format.
						`tftp://inetboot.sun4u`
						For the WAN boot file, the format is
						`http://`*host.domain/path-to-file*
						This option can be used to override `BootFile` and `siaddr` settings in order to retrieve a standalone boot file. Supported protocols: tftp (inetboot), http (wanboot). For example, use the following format. `tftp://inetboot.sun4u`

`SHTTPproxy`	17	ASCII text	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						IP address and port number of the proxy server that is used on your network. This option is needed only when a client is booting across a WAN, and the local network uses a proxy server. For example, use the following format: `198.162.10.5:8080`

The following options are not currently used by the Solaris client startup scripts. You can use them only if you edit the startup scripts.						

`SswapIP4`	5	IP address	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						IP address of swap server

`SswapPTH`	6	ASCII text	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						Path to the client's swap file on the swap server

TABLE 7–4 Values for Creating Vendor Category Options for Solaris Clients *(Continued)*

Name	Code	Data Type	Granularity	Maximum	Vendor Client Classes *	Description

`Stz`	8	ASCII text	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						Time zone for client

`Sterm`	15	ASCII text	1	0	`SUNW.Sun-Blade-1000,`
					`SUNW.Sun-Fire-880,`
					`SUNW.i86pc`
						Terminal type

When you have created the options, you can create macros that include those options. The following table lists sample macros you can create to support Solaris installation for clients.

TABLE 7–5 Sample Macros to Support Network Installation Clients

Macro Name	Contains These Options and Macros
`Solaris`	`SrootIP4, SrootNM, SinstIP4, SinstNM`
`sparc`	`SrootPTH, SinstPTH`
`sun4u`	`Solaris` and `sparc` macros
`i86pc`	`Solaris` macro, `SrootPTH, SinstPTH, SbootFIL`
`SUNW.i86pc`	`i86pc` macro
`SUNW.Sun-Blade-1000`	`sun4u` macro, `SbootFIL`
`SUNW.Sun-Fire-880`	`sun4u` macro, `SbootFIL`
xxx.xxx.xxx.xxx network address macros	`BootSrvA` option could be added to existing network address macros. The value of `BootSrvA` should indicate the `tftboot` server.

The macro names that are listed in the previous table match the Vendor client classes of the clients that must install from the network. These names are examples of clients you might have on your network. See "Working With DHCP Options (Task Map)" in *System Administration Guide: IP Services* for information about determining a client's vendor client class.

You can create these options and macros by using the following methods.

- Write a script that creates the options and macros by using the `dhtadm` command. See "Writing a Script That Uses `dhtadm` to Create Options and Macros" on page 91 for information bout how to write scripts that create these options and macros.

- Create the options and macros in DHCP Manager. See “Using DHCP Manager to Create Install Options and Macros” on page 93 for instructions about how to create options and macros in DHCP Manager.

Writing a Script That Uses `dhtadm` to Create Options and Macros

You can create a Korn shell script by adapting the example in Example 7–13 to create all the options listed in Table 7–4 and some useful macros. Be sure to change all IP addresses and values contained in quotes to the correct IP addresses, server names, and paths for your network. You should also edit the `Vendor=` key to indicate the class of clients you have. Use the information that `add_install_client -d` reports to obtain the data that you need to adapt the script.

EXAMPLE 7–13 Sample Script to Support Network Installation

```
# Load the Solaris vendor specific options. We’ll start out supporting
# the Sun-Blade-1000, Sun-Fire-880, and i86 platforms. Changing -A to -M would replace
# the current values, rather than add them.
dhtadm -A -s SrootOpt -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,1,ASCII,1,0’
dhtadm -A -s SrootIP4 -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,2,IP,1,1’
dhtadm -A -s SrootNM -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,3,ASCII,1,0’
dhtadm -A -s SrootPTH -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,4,ASCII,1,0’
dhtadm -A -s SswapIP4 -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,5,IP,1,0’
dhtadm -A -s SswapPTH -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,6,ASCII,1,0’
dhtadm -A -s SbootFIL -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,7,ASCII,1,0’
dhtadm -A -s Stz -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,8,ASCII,1,0’
dhtadm -A -s SbootRS -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,9,NUMBER,2,1’
dhtadm -A -s SinstIP4 -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,10,IP,1,1’
dhtadm -A -s SinstNM -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,11,ASCII,1,0’
dhtadm -A -s SinstPTH -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,12,ASCII,1,0’
dhtadm -A -s SsysidCF -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,13,ASCII,1,0’
dhtadm -A -s SjumpsCF -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,14,ASCII,1,0’
dhtadm -A -s Sterm -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,15,ASCII,1,0’
dhtadm -A -s SbootURI -d \
’Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,16,ASCII,1,0’
```

```
dhtadm -A -s SHTTPproxy -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,17,ASCII,1,0'
# Load some useful Macro definitions.
# Define all Solaris-generic options under this macro named Solaris.
dhtadm -A -m Solaris -d \
':SrootIP4=10.21.0.2:SrootNM="blue2":SinstIP4=10.21.0.2:SinstNM="red5":'
# Define all sparc-platform specific options under this macro named sparc.
dhtadm -A -m sparc -d \
':SrootPTH="/export/sparc/root":SinstPTH="/export/sparc/install":'
# Define all sun4u architecture-specific options under this macro named sun4u.
#  (Includes Solaris and sparc macros.)
dhtadm -A -m sun4u -d ':Include=Solaris:Include=sparc:'
# Solaris on IA32-platform-specific parameters are under this macro named i86pc.
dhtadm -A -m i86pc -d \
':Include=Solaris:SrootPTH="/export/i86pc/root":SinstPTH="/export/i86pc/install"\
:SbootFIL="/platform/i86pc/kernel/unix":'
# Solaris on IA32 machines are identified by the "SUNW.i86pc" class. All
# clients identifying themselves as members of this class will see these
# parameters in the macro called SUNW.i86pc, which includes the i86pc macro.
dhtadm -A -m SUNW.i86pc -d ':Include=i86pc:'
# Sun-Blade-1000 platforms identify themselves as part of the
# "SUNW.Sun-Blade-1000" class.
# All clients identifying themselves as members of this class
#  will see these parameters.
dhtadm -A -m SUNW.Sun-Blade-1000 -d \
':SbootFIL="/platform/sun4u/kernel/sparcv9/unix":\
Include=sun4u:'
# Sun-Fire-880 platforms identify themselves as part of the "SUNW.Sun-Fire-880" class.
# All clients identifying themselves as members of this class will see these parameters.
dhtadm -A -m SUNW.Sun-Fire-880 -d \
':SbootFIL="/platform/sun4u/kernel/sparcv9/unix":Include=sun4u:'
# Add our boot server IP to each of the network macros for our topology served by our
# DHCP server. Our boot server happens to be the same machine running our DHCP server.
dhtadm -M -m 10.20.64.64 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.128 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.21.0.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.22.0.0    -e BootSrvA=10.21.0.2
# Make sure we return host names to our clients.
dhtadm -M -m DHCP-servername -e Hostname=_NULL_VALUE_
# The client with this MAC address is a diskless client. Override the root settings
# which at the network scope setup for Install with our client's root directory.
dhtadm -A -m 0800201AC25E -d \
':SrootIP4=10.23.128.2:SrootNM="orange-svr-2":SrootPTH="/export/root/10.23.128.12":'
```

As superuser, execute `dhtadm` in batch mode. Specify the name of the script to add the options and macros to your `dhcptab`. For example, if your script is named `netinstalloptions`, type the following command.

```
# dhtadm -B netinstalloptions
```

Clients that have vendor client classes that are listed in the Vendor= string can now use DHCP to install over the network.

For more information about how to use the dhtadm command, see dhtadm(1M). For more information about the dhcptab file, see dhcptab(4).

Using DHCP Manager to Create Install Options and Macros

You can use DHCP Manager to create the options that are listed in Table 7–4 and the macros that are listed in Table 7–5.

▼ How to Create Options to Support Solaris Installation (DHCP Manager)

This procedure assumes that you have already configured your DHCP server. If you have not configured your DHCP server, see “Planning for DHCP Service (Task)” in *System Administration Guide: IP Services*.

1. **Become superuser on the DHCP server system.**
2. **Start the DHCP Manager.**

   ```
   # /usr/sadm/admin/bin/dhcpmgr &
   ```

   The DHCP Manager window is displayed.
3. **Select the Options tab in DHCP Manager.**
4. **Choose Create from the Edit menu.**

   The Create Option dialog box opens.
5. **Type the option name for the first option, then type values appropriate for that option.**

   Use Table 7–4 to check the option names and values for options you must create. Notice that the vendor client classes are only suggested values. You should create classes to indicate the actual client types that need to obtain Solaris installation parameters from the DHCP service. See “Working With DHCP Options (Task Map)” in *System Administration Guide: IP Services* for information about how to determine a client’s vendor client class.
6. **Click OK when you have entered all the values.**
7. **In the Options tab, select the option you just created.**
8. **Select Duplicate from the Edit menu.**

The Duplicate Option dialog box opens.

9. **Type the name of another option, then modify other values appropriately.**

   The values for code, data type, granularity, and maximum are most likely to need modification. See Table 7–4 for the values.

10. **Repeat Step 7 through Step 9 until you have created all the options.**

    You can now create macros to pass the options to network installation clients, as explained in the following procedure.

Note – You do not need to add these options to a Solaris client's `/etc/dhcp/inittab` file because they are already included in that file.

▼ How to Create Macros to Support Solaris Installation (DHCP Manager)

This procedure assumes that you have already configured your DHCP server. If you have not configured your DHCP server, see "Planning for DHCP Service (Task)" in *System Administration Guide: IP Services*.

1. **Select the Macros tab in DHCP Manager.**

2. **Choose Create from the Edit menu.**

   The Create Macro dialog box opens.

3. **Type the name of a macro.**

   See Table 7–5 for macro names you might use.

4. **Click the Select button.**

   The Select Option dialog box opens.

5. **Select Vendor in the Category list.**

   The Vendor options you created are listed.

6. **Select an option you want to add to the macro and click OK.**

7. **Type a value for the option.**

   See Table 7–4 for the option's data type and refer to the information that `add_install_client -d` reports.

8. **Repeat Step 6 through Step 7 for each option you want to include.**

   To include another macro, type **`Include`** as the option name and type the macro name as the option value.

9. **Click OK when the macro is complete.**

SPARC: Preconfiguring Power Management Information

You can use the *Power Management* software that is provided in the Solaris environment to automatically save the state of a system and turn it off after it is idle for 30 minutes. When you install the Solaris 9 software on a system that complies with version 2 of the EPA's Energy Star guidelines, for example a sun4u system, the Power Management software is installed by default. During a Solaris Web Start installation, the installation program prompts you to enable or disable the Power Management software. The `suninstall` installation program prompts you to enable or disable the Power Management software after the installation is complete and the system reboots.

Note – If your system has Energy Star version 3 or later, you are not prompted for this information.

If you are performing interactive installations, you cannot preconfigure the Power Management information and avoid the prompt. However, by using a custom JumpStart installation, you can preconfigure the Power Management information by using a finish script to create an `/autoshutdown` or `/noautoshutdown` file on the system. When the system reboots, the `/autoshutdown` file enables Power Management and the `/noautoshutdown` file disables Power Management.

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

```
touch /a/autoshutdown
```

Finish scripts are described in "Creating Finish Scripts" on page 295.

CHAPTER **8**

Upgrading the Solaris Operating Environment (Planning)

This chapter provides specific information and instructions about tasks that you must perform before you upgrade to the Solaris operating environment.

- "Upgrading (Overview)" on page 97
- "Using Solaris Live Upgrade" on page 99
- "Using Custom JumpStart to Upgrade" on page 99
- "Upgrading With Disk Space Reallocation" on page 100
- "Backing Up Systems Before Upgrading" on page 101

Upgrading (Overview)

An upgrade merges the new version of the Solaris operating environment with the existing files on the system's disk. An upgrade saves as many modifications as possible that you have made to the previous version of the Solaris operating environment.

You can upgrade any system that is running the Solaris 2.6, Solaris 7, or Solaris 8. Type the following command to see the version of Solaris software that is running on your system:

```
$ uname -a
```

You can upgrade the Solaris operating environment by using the following installation methods.

Note – Use the `smosservice patch` to upgrade diskless clients. For detailed instructions, refer to *System Administration Guide: Basic Administration* or to `smosservice`(1M).

If you are already running the Solaris 9 operating environment and have installed individual patches, upgrading to a Solaris 9 Update release causes the following:

- Any patches that were supplied as part of the Solaris 9 Update release are reapplied to your system. You cannot back out these patches.
- Any patches that were previously installed on your system and are not included in the Solaris 9 Update release are removed.

You can use the Patch Analyzer to determine which patches, if any, will be removed by upgrading to the Solaris 9 Update release. For detailed instructions about using the Patch Analyzer, refer to "Upgrading to a Solaris Update Release" on page 661.

Solaris Upgrade Methods

TABLE 8–1 SPARC: Solaris Upgrade Methods

Platform	Current Solaris Operating Environment	Solaris Upgrade Methods
SPARC systems	Solaris 2.6, Solaris 7, Solaris 8, Solaris 9	■ The Solaris Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart method ■ Solaris Live Upgrade

TABLE 8–2 x86: Solaris Upgrade Methods

Current Solaris Operating Environment	Solaris Upgrade Methods
Solaris 2.6	Installing from DVD media or a net installation image: ■ Solaris Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart method Installing from CD media: ■ Solaris `suninstall` program ■ Custom JumpStart method
Solaris 7	Installing from DVD media or a net installation image: ■ Solaris Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart method ■ Solaris Live Upgrade Installing from CD media: ■ Solaris `suninstall` program ■ Custom JumpStart method ■ Solaris Live Upgrade

TABLE 8–2 x86: Solaris Upgrade Methods *(Continued)*

Current Solaris Operating Environment	Solaris Upgrade Methods
Solaris 8, Solaris 9	Installing from DVD or CD media or a net installation image: ■ Solaris Web Start program ■ Solaris `suninstall` program ■ Custom JumpStart method ■ Solaris Live Upgrade

Note – For limitations on upgrading using Solaris Live Upgrade, see "Solaris Live Upgrade Requirements" on page 395.

You cannot upgrade your system to a software group that is not installed on the system. For example, if you previously installed the End User Solaris Software Group on your system, you cannot use the upgrade option to upgrade to the Developer Solaris Software Group. However, during the upgrade you can add software to the system that is not part of the currently installed software group.

Using Solaris Live Upgrade

Solaris Live Upgrade enables an upgrade on a duplicate, inactive operating environment, which reduces the downtime of an operating system upgrade.

For instructions on how to plan for and use Solaris Live Upgrade, see Chapter 32.

Using Custom JumpStart to Upgrade

You can use the custom JumpStart installation method to upgrade. In the custom JumpStart profile, specify `install_type upgrade`.

You must test the custom JumpStart profile against the system's disk configuration and currently installed software before you upgrade. Use the `pfinstall -D` command on the system that you are upgrading to test the profile. You cannot test an upgrade profile by using a disk configuration file. For more information about testing the upgrade option, refer to "Testing a Profile" on page 286.

You can use JumpStart to update a clone system with a Solaris Flash differential archive. For an overview of Solaris Flash archives, see Chapter 20.

Upgrading With Disk Space Reallocation

The upgrade option in the Solaris Web Start installation method, the Solaris `suninstall` program, and the custom JumpStart program provide the ability to reallocate disk space. You can reallocate disk space if the current file systems do not have enough space for the upgrade. File systems might need more space for the upgrade for the following reasons:

- The Solaris software group that is currently installed on the system contains new software in the new release. Any new software that is included in a software group is automatically selected to be installed during the upgrade.
- The size of the existing software on the system has increased in the new release.

The auto-layout feature attempts to reallocate the disk space to accommodate the new size requirements of the file system. Initially, auto-layout attempts to reallocate space, based on a set of default constraints. If auto-layout cannot reallocate space, you must change the constraints on the file systems.

Note – Auto-layout does not have the ability to grow file systems. Auto-layout reallocates space by backing up required files on the file systems that need to change, repartitioning the disks on the basis of the file system changes, and restoring the backup files before the upgrade happens.

- If you are using the Solaris Web Start program, and auto-layout cannot determine how to reallocate the disk space, you must use the Solaris `suninstall` program or the custom JumpStart program to upgrade.
- If you are using the Solaris `suninstall` program, and auto-layout cannot determine how to reallocate disk space, you must specify the file systems that can be moved or changed and run auto-layout again.
- If you are using the custom JumpStart method to upgrade and you create an upgrade profile, disk space might be a concern. If the current file systems do not contain enough disk space for the upgrade, you can use the `backup_media` and `layout_constraint` keywords to reallocate disk space. For an example of how to use the `backup_media` and `layout_constraint` keywords in a profile, refer to Example 26–5.

Backing Up Systems Before Upgrading

Back up existing file systems before you upgrade to the Solaris operating environment. If you copy file systems to removable media, such as tape, you can safeguard against data loss, damage, or corruption. For detailed instructions to back up your system, refer to *System Administration Guide: Devices and File Systems*.

CHAPTER 9

Creating RAID-1 Volumes (Mirrors) During Installation (Topics)

This section provides an overview of Solaris Volume Manager components. The section also describes the guidelines and requirements that are necessary to mirror file systems during your installation or uprade.

Chapter 10	Provides an overview of the Solaris Volume Manager components that are necessary to mirror file systems.
Chapter 11	Describes the guidelines and requirements to create mirrored file systems during your installation or upgrade.

CHAPTER 10

Creating RAID-1 Volumes (Mirrors) During Installation (Overview)

This section discusses the advantages of creating mirrored file systems. The section also describes the Solaris Volume Manager components that are required to create mirrored file systems.

This chapter describes the following topics.

- "Why Mirror?" on page 105
- "How Mirroring Works" on page 106
- "Overview of Mirror Components" on page 108
- "Sample Layout for Mirrored File Systems" on page 111

For additional information about how to create mirrored file systems with Solaris Live Upgrade, see "General Guidelines for Creating Mirrored File Systems" on page 402.

For additional information about how to create mirrored file systems with the custom JumpStart installation method, see "`filesys` Profile Keyword (Creating Mirrored File Systems)" on page 358 and "`metadb` Profile Keyword (Creating State Database Replicas)" on page 364.

Why Mirror?

During the installation or upgrade, you can create mirrored file systems to duplicate your system data over multiple physical disks. By duplicating your data over separate disks, you can protect your data from disk corruption or a disk failure.

The Solaris custom JumpStart and Solaris Live Upgrade installation methods use the Solaris Volume Manager technology to create a mirrored file system. Solaris Volume Manager provides a powerful way to reliably manage your disks and data by using volumes. Solaris Volume Manager enables concatenations, stripes, and other complex configurations. The custom JumpStart and Solaris Live Upgrade installation methods enable a subset of these tasks, such as creating a RAID-1 volume for the root (/) file system. You can create mirrored file systems during your installation or upgrade, eliminating the need to create the mirrored file system after the installation.

Note – The custom JumpStart and Solaris Live Upgrade installation methods only support the creation of RAID-0 and RAID-1 volumes. Other Solaris Volume Manager components, such as RAID-5 volumes, are not supported.

The custom JumpStart installation method supports the creation of mirrored file systems during an initial installation only. Solaris Live Upgrade supports the creation of mirrored file systems during an upgrade.

For detailed information about Solaris Volume Manager software and components, see *Solaris Volume Manager Administration Guide*.

How Mirroring Works

Solaris Volume Manager uses virtual disks to manage physical disks and their associated data. In Solaris Volume Manager, a virtual disk is called a *volume*. A *volume* is a name for a group of physical slices that appear to the system as a single, logical device. Volumes are actually pseudo, or virtual, devices in standard UNIX® terms.

A volume is functionally identical to a physical disk in the view of an application or a file system (such as UFS). Solaris Volume Manager converts I/O requests directed at a volume into I/O requests to the underlying member disks.

Solaris Volume Manager volumes are built from slices (disk partitions) or from other Solaris Volume Manager volumes.

You use volumes to increase performance and data availability. In some instances, volumes can also increase I/O performance. Functionally, volumes behave the same way as slices. Because volumes look like slices, they are transparent to end users, applications, and file systems. Like physical devices, you can use Solaris Volume Manager software to access volumes through block or raw device names. The volume name changes, depending on whether the block or raw device is used.

The custom JumpStart installation method and Solaris Live Upgrade support the use of block devices to create mirrored file systems. See "RAID Volume Name Requirements and Guidelines for Custom JumpStart and Solaris Live Upgrade" on page 116 for details about volume names.

When you create a mirrored file system, you create RAID-0 volumes (single-slice concatenations) and RAID-1 volumes (mirrors.) Solaris Volume Manager duplicates data on the concatentations, (submirrors), and treats the submirrors as one mirror volume.

Figure 10–1 shows a mirror that duplicates the root (/) file system over two physical disks.

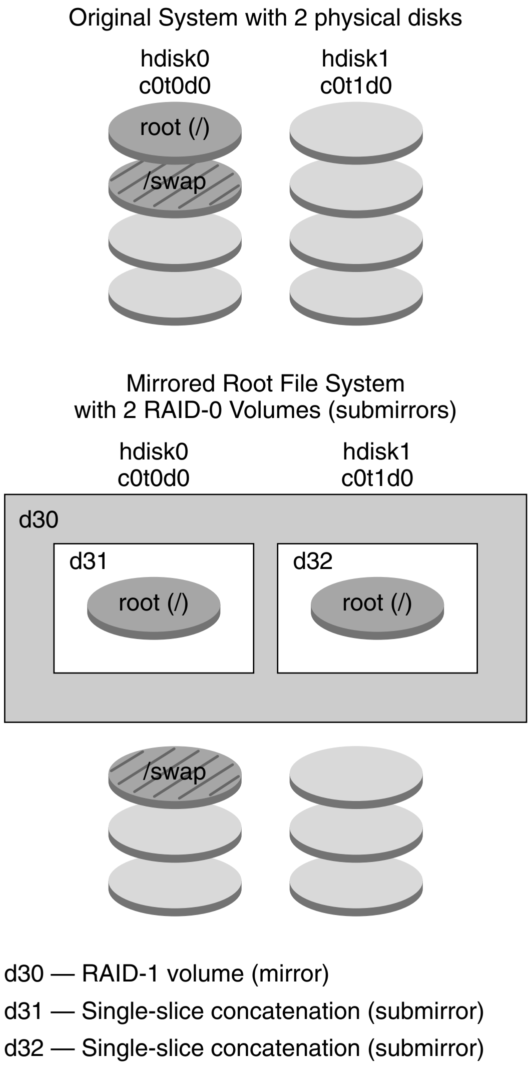

Original System with 2 physical disks
hdisk0
c0t0d0
hdisk1
c0t1d0
root (/)
/swap
Mirrored Root File System
with 2 RAID-0 Volumes (submirrors)
hdisk0
c0t0d0
hdisk1
c0t1d0
d30
d31
root (/)
d32
root (/)
/swap
d30 — RAID-1 volume (mirror)
d31 — Single-slice concatenation (submirror)
d32 — Single-slice concatenation (submirror)

FIGURE 10–1 Mirroring the Root File System on Two Disks

Figure 10–1 shows a system with the following configuration.

- The root file system (/) on hdisk0 is included in the single-slice concatenation that is named d31.
- A single-slice concatenation that is named d32 is created on the hard disk that is named hdisk1.
- The mirror that is named d30 consists of the submirrors that are named d31 and d32.
- The mirror duplicates the data in the root file system on both submirrors.

Overview of Mirror Components

The custom JumpStart installation method and Solaris Live Upgrade enable you to create the following components that are required to mirror a file system.

- State database and state database replicas (metadbs)
- Single-slice concatenations (submirrors)
- RAID-1 volumes (mirrors)

This section briefly describes each of these components. For complete information about these components, see *Solaris Volume Manager Administration Guide*.

State Database and State Database Replicas

The *state database* is a database that stores information on a physical disk about the state of your Solaris Volume Manager configuration. The state database records and tracks changes that are made to your configuration. Solaris Volume Manager automatically updates the state database when a configuration or state change occurs. Creating a new volume is an example of a configuration change. A submirror failure is an example of a state change.

The state database is actually a collection of multiple, replicated database copies. Each copy, referred to as a *state database replica*, ensures that the data in the database is always valid. Having copies of the state database protects against data loss from single points of failure. The state database tracks the location and status of all known state database replicas.

Solaris Volume Manager cannot operate until you have created the state database and its state database replicas. A Solaris Volume Manager configuration must have an operating state database.

When you set up your configuration, you can locate the state database replicas on either of the following:

- Dedicated slices

- (Solaris Live Upgrade only) Slices that will later become part of volumes

You can keep more than one copy of a state database on one slice. However, you might make the system more vulnerable to a single point of failure by placing state database replicas on a single slice.

The state database replicas ensure that the data in the state database is always valid. When the state database is updated, each state database replica is also updated. The updates occur one at a time to protect against corruption of all updates if the system crashes.

If your system loses a state database replica, Solaris Volume Manager must identify which state database replicas still contain valid data. Solaris Volume Manager determines this information by using a *majority consensus algorithm*. This algorithm requires that a majority (half + 1) of the state database replicas be available and in agreement before any of them are considered valid. Because of this majority consensus algorithm, you must create at least three state database replicas when you set up your disk configuration. A consensus can be reached if at least two of the three state database replicas are available.

Each state database replica occupies 4 Mbytes (8192 disk sectors) of disk storage by default. Replicas can be stored on the following devices:

- A dedicated local disk slice
- (Solaris Live Upgrade only) A local slice that will be part of a volume
- (Solaris Live Upgrade only) A local slice that will be part of a UFS logging device

Replicas cannot be stored on the root (`/`), `swap`, or `/usr` slices, or on slices that contain existing file systems or data. After the replicas have been stored, volumes or file systems can be placed on the same slice.

For planning information about state database and state database replica requirements, see “State Database Replicas Guidelines and Requirements” on page 114.

For detailed information about the state database and state database replicas, see *Solaris Volume Manager Administration Guide*.

RAID-0 Volumes (Concatenations)

The custom JumpStart and Solaris Live Upgrade installation methods enable you to create RAID-0 volumes. A RAID-0 volume single-slice concatenation is a volume whose data is organized serially and adjacently across components, forming one logical storage unit. The custom JumpStart installation method and Solaris Live Upgrade do not enable you to create stripes or other complex Solaris Volume Manager volumes.

During the installation or upgrade, you can create RAID-1 volumes (mirrors) and attach RAID-0 volumes to these mirrors. The RAID-0 volumes that are *mirrored* are called *submirrors*. A mirror is made of one or more RAID-0 volumes. After the installation, you can manage the data on separate RAID-0 submirror volumes by administering the RAID-1 mirror volume through the Solaris Volume Manager software.

The custom JumpStart installation method enables you to create a mirror that consists of up to two submirrors. Solaris Live Upgrade enables you to create a mirror that consists of up to three submirrors. Practically, a two-way mirror is usually sufficient. A third submirror enables you to make online backups without losing data redundancy while one submirror is offline for the backup.

For planning information about RAID–0 volume requirements, see "Mirror and Submirror Requirements and Guidelines" on page 115.

For detailed information about RAID-0 volumes, see *Solaris Volume Manager Administration Guide*.

RAID-1 Volumes (Mirrors)

A RAID-1 volume, or *mirror*, is a volume that maintains identical copies of the data in RAID-0 volumes (single-slice concatenations.) Mirroring requires an investment in disks. You need at least twice as much disk space as the amount of data you have to mirror. Because Solaris Volume Manager software must write to all submirrors, mirroring can also increase the time that is required for write requests to be written to disk.

With RAID-1 volumes, data can be read from both RAID-0 volumes simultaneously (either volume can service any request), providing improved performance. If one physical disk fails, you can continue to use the mirror with no loss in performance or loss of data.

After you configure a mirror, the mirror can be used just as if it were a physical slice.

You can mirror any file system, including existing file systems. You can also use a mirror for any application, such as a database.

For planning information about RAID–1 volume requirements, see "Mirror and Submirror Requirements and Guidelines" on page 115.

For detailed information about RAID-1 volumes, see *Solaris Volume Manager Administration Guide*.

Sample Layout for Mirrored File Systems

The following figure shows a mirror that duplicates the root file system (/) over two physical disks. State database replicas (metadbs) are placed on both disks.

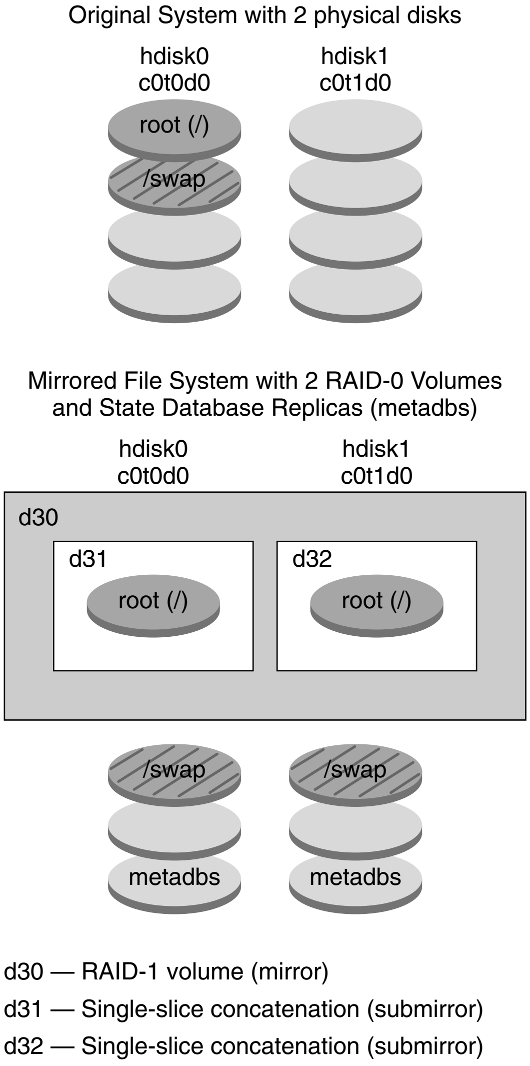

Original System with 2 physical disks
hdisk0
c0t0d0
hdisk1
c0t1d0
root (/)
/swap
Mirrored File System with 2 RAID-0 Volumes
and State Database Replicas (metadbs)
hdisk0
c0t0d0
hdisk1
c0t1d0
d30
d31
root (/)
d32
root (/)
/swap
/swap
metadbs
metadbs
d30 — RAID-1 volume (mirror)
d31 — Single-slice concatenation (submirror)
d32 — Single-slice concatenation (submirror)

FIGURE 10–2 Sample Layout for a Mirrored Root File System

Figure 10–2 shows a system with the following configuration.

- The root file system (/) on hdisk0 is included in the single-slice concatenation that is named d31.
- A single-slice concatenation that is named d32 is created on the hard disk that is named hdisk1.
- The mirror that is named d30 consists of the submirrors that are named d31 and d32.
- The mirror duplicates the data in the root file system on both submirrors.
- State database replicas are created on slices both hdisk0 and hdisk1.

For an example profile that uses the custom JumpStart installation method to create this configuration, see Example 26–10.

For instructions on how to create mirrored file systems with Solaris Live Upgrade, see “To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)” on page 435.

CHAPTER 11

Creating RAID-1 Volumes (Mirrors) During Installation (Planning)

This chapter describes the requirements and guidelines that are necessary to create mirrored file systems with the custom JumpStart or Solaris Live Upgrade installation methods.

This chapter describes the following topics.

- "System Requirement" on page 113
- "State Database Replicas Guidelines and Requirements" on page 114
- "Mirror and Submirror Requirements and Guidelines" on page 115
- "How Booting Into Single-User Mode Affects RAID-1 Volumes" on page 119

For additional information about planning to create mirrored file systems with the Solaris Live Upgrade installation method, see "General Guidelines for Creating Mirrored File Systems" on page 402.

For instructions about how to create mirrored file systems with the custom JumpStart installation method, see "`filesys` Profile Keyword (Creating Mirrored File Systems)" on page 358 and "`metadb` Profile Keyword (Creating State Database Replicas)" on page 364.

System Requirement

To create mirrored file systems on specific slices, the disks that you plan to use for mirroring must be directly attached and available to the system during the installation.

State Database Replicas Guidelines and Requirements

You should distribute state database replicas across slices, drives, and controllers, to avoid single points of failure. You want a majority of replicas to survive a single component failure. If you lose a replica, when a device fails, for example, the failure might cause problems with running Solaris Volume Manager software or when rebooting the system. Solaris Volume Manager software requires at least half of the replicas to be available to run, but a majority (half plus one) to reboot into multiuser mode.

For detailed instructions about creating and administering state database replicas, see *Solaris Volume Manager Administration Guide*.

Selecting Slices for State Database Replicas

Before selecting slices for state database replicas, consider the following guidelines and recommendations.

- You should create state database replicas on a dedicated slice of at least 4 Mbytes per replica. If necessary, you could create state database replicas on a slice that is to be used as part of a RAID-0 or RAID-1 volume. You must create the replicas before you add the slice to the volume.
- By default, the size of a state database replica is 4 Mbytes or 8192 disk blocks. Because your disk slices might not be that small, you can resize a slice to hold the state database replica. For information about resizing a slice, see "Administering Disks (Tasks)" in *System Administration Guide: Devices and File Systems*.
- You can create state database replicas on slices that are not in use. The part of a slice that is reserved for the state database replica should not be used for any other purpose.
- You cannot create state database replicas on existing file systems, or the root (`/`), `/usr`, and `swap` file systems. If necessary, you can create a new slice (provided a slice name is available) by allocating space from `swap` and then put state database replicas on that new slice.
- When a state database replica is placed on a slice that becomes part of a volume, the capacity of the volume is reduced by the space that is occupied by the replica or replicas. The space that is used by a replica is rounded up to the next cylinder boundary and this space is skipped by the volume.

Choosing the Number of State Database Replicas

Before choosing the number of state database replicas, consider the following guidelines.

- A minimum of 3 state database replicas are recommended, up to a maximum of 50 replicas per Solaris Volume Manager disk set. The following guidelines are recommended:
  - For a system with only a single drive: put all three replicas in one slice.
  - For a system with two to four drives: put two replicas on each drive.
  - For a system with five or more drives: put one replica on each drive.
- Additional state database replicas can improve the mirror's performance. Generally, you need to add two replicas for each mirror you add to the system.
- If you have a RAID-1 volume that is to be used for small-sized random I/O (for example, for a database), consider your number of replicas. For best performance, ensure that you have at least two extra replicas per RAID-1 volume on slices (and preferably on disks and controllers) that are unconnected to the RAID-1 volume.

Distributing State Database Replicas Across Controllers

If multiple controllers exist, replicas should be distributed as evenly as possible across all controllers. This strategy provides redundancy if a controller fails and also helps balance the load. If multiple disks exist on a controller, at least two of the disks on each controller should store a replica.

Mirror and Submirror Requirements and Guidelines

When you are working with RAID-1 volumes (mirrors) and RAID-0 volumes (single-slice concatenations), consider the following guidelines.

Custom JumpStart and Solaris Live Upgrade Guidelines

The custom JumpStart installation method and Solaris Live Upgrade support a subset of the features that are available in the Solaris Volume Manager software. When you create mirrored file systems with these installation programs, consider the following guidelines.

- The term RAID-0 volume can refer to disk stripes or disk concatenations. The custom JumpStart and Solaris Live Upgrade installation methods only enable you to create single-slice concatenations. You cannot create RAID-0 stripe volumes during the installation or upgrade.
- The custom JumpStart installation method enables you to create up to two submirrors for each mirror. The Solaris Live Upgrade installation method enables you to create up to three submirrors for each mirror. Two submirrors usually provide sufficient data redundancy for most applications, and the disk drive costs are less expensive. Three submirrors enable you to take a submirror offline and perform a backup while maintaining the two remaining submirrors for continued data redundancy.
- If you create mirrored file systems with the custom JumpStart installation method, you do not need to create the file systems that you are mirroring before you create the mirror.

RAID Volume Name Requirements and Guidelines for Custom JumpStart and Solaris Live Upgrade

Observe the following rules when assigning names for volumes.

- Use a naming method that maps the slice number and disk number to volume numbers.
- Volume names must begin with the letter `d` followed by a number, for example, `d0`.
- Solaris Volume Manager has 128 default volume names from 0–127. The following list shows some example volume names.
  - Device `/dev/md/dsk/d0` — block volume `d0`
  - Device `/dev/md/dsk/d1` — block volume `d1`
- Use ranges for each particular type of volume. For example, assign numbers 0–20 for RAID-1 volumes, and 21–40 for RAID-0 volumes.
- Instead of specifying the full volume name, such as `/dev/md/dsk/d1`, you can often use an abbreviated volume name, such as `d1`.

RAID Volume Naming Conventions for Solaris Live Upgrade

You can abbreviate the names of physical disk slices and Solaris Volume Manager volumes. The abbreviation is the shortest name that uniquely identifies a device. Examples follow.

- A Solaris Volume Manager volume can be identified by its d*num* designation, so that, for example, `/dev/md/dsk/d10` becomes simply `d10`.
- If a system has a single controller and multiple disks, you might use `t0d0s0`, but with multiple controllers use `c0t0d0s0`.

When you use the custom Solaris Live Upgrade to create RAID-1 volumes (mirrors) and RAID-0 volumes (submirrors), you can let the software detect and assign volume names, or you can assign the names. If you let the software detect the names, the software assigns the first mirror or submirror name that is available. If you assign mirror names, assign names ending in zero so that the installation can use the names ending in 1 and 2 for submirrors. If you assign submirror names, assign names ending in 1 or 2. If you assign numbers incorrectly, the mirror might not be created. For example, if you specify a mirror name with a number that ends in 1 or 2 (`d1` or `d2`), Solaris Live Upgrade fails to create the mirror if the mirror name is a duplicate of a submirror's name.

In this example, Solaris Live Upgrade assigns the volume names. The RAID-1 volumes `d0` and `d1` are the only volumes in use. For the mirror `d10`, Solaris Live Upgrade picks `d2` for the submirror for the device `c0t0d0s0` and `d3` for the submirror for the device `c1t0d0s0`.

```
lucreate -n newbe -m /:d10:mirror,ufs -m /:c0t0d0s0:attach -m  \
/:c1t0d0s0:attach
```

In this example, the volume names are assigned in the command. For the mirror `d10`, `d11` is the name for the submirror for the device `c0t0d0s0` and `d12` is the name for the submirror for the device `c1t0d0s0`.

```
lucreate -n newbe -m /:d10:mirror,ufs -m /:c0t0d0s0,d11:attach -m \
/:c1t0d0s0,d12:attach
```

For detailed information about Solaris Volume Manager naming requirements, see *Solaris Volume Manager Administration Guide*.

RAID Volume Naming Conventions for Custom JumpStart

When you use the custom JumpStart installation method to create RAID-1 volumes (mirrors) and RAID-0 volumes (submirrors), you can let the software detect and assign volume names to mirrors, or you can assign the names in the profile. If you let the software detect the names, the software assigns the first volume number that is available. If you assign names in the profile, assign mirror names ending in zero so that the installation can use the names ending in 1 and 2 for submirrors. If you assign numbers incorrectly, the mirror might not be created. For example, if you specify a mirror name with a number that ends in 1 or 2 (`d1` or `d2`), Jumpstart fails to create the mirror if the mirror name is a duplicate of a submirror's name. In the following profile example, the mirror is assigned the first volume numbers that are available. If the next available mirror ending in zero is `d10`, then the names `d11` and `d12` are assigned to the submirrors.

```
filesys                 mirror c0t0d0s1  /
```

In the following profile example, the mirror number is assigned in the profile as d30. The submirror names are assigned by the software based on the mirror number and the first available submirrors. In this example, the submirrors are named d31 and d32.

```
filesys                 mirror:d30 c0t1d0s0 c0t0d0s0  /
```

For detailed information about Solaris Volume Manager naming requirements, see *Solaris Volume Manager Administration Guide*.

Guidelines for Selecting Disks and Controllers

When you choose the disks and controllers that you want to use to mirror a file system, consider the following guidelines.

- Use components that are on different controllers to increase the number of simultaneous reads and writes that can be performed.
- Keep the slices of different submirrors on different disks and controllers. Data protection is diminished considerably if slices of two or more submirrors of the same mirror are on the same disk.
- Organize submirrors across separate controllers, because controllers and associated cables tend to fail more often than disks. This practice also improves mirror performance.
- Use the same type of disks and controllers in a single mirror. Particularly in old SCSI storage devices, different models or brands of disk or controller can have widely varying performance. Mixing the different performance levels in a single mirror can cause performance to degrade significantly.

Guidelines for Selecting Slices

When you choose the slices that you want to use to mirror a file system, consider the following guidelines.

- Any file system, including root (/), `swap`, and `/usr`, can use a mirror. Any application, such as a database, also can use a mirror.
- Make sure that your submirror slices are of equal size. Submirrors of different sizes result in unused disk space.
- If you have a mirrored file system in which the first submirror attached does not start on cylinder 0, all additional submirrors you attach must also not start on cylinder 0. If you attempt to attach a submirror starting on cylinder 0 to a mirror in which the original submirror does not start on cylinder 0, the following error message is displayed:

  ```
  can’t attach
  labeled submirror to an unlabeled mirror
  ```

  You must ensure that all submirrors you plan to attach to a mirror either all start on cylinder 0, or that none of them start on cylinder 0. Starting cylinders do not have to be identical across all submirrors, but all submirrors must either include or not include cylinder 0.

How Booting Into Single-User Mode Affects RAID-1 Volumes

If a system with mirrors for root (/), /usr, and swap is booted into single-user mode, the system indicates that these mirrors are in need of maintenance. When you view these mirrors with the metastat command, these mirrors, and possibly all mirrors on the system, appear in the "Needing Maintenance" state.

Though this situation appears to be potentially dangerous, do not be concerned. The metasync -r command, which normally occurs during boot to resynchronize mirrors, is interrupted when the system is booted into single-user mode. After the system is rebooted, the metasync -r command runs and resynchronizes all mirrors.

If this interruption is a concern, run the metasync -r command manually.

For more information on the metasync, see the metasync(1M) man page, and *Solaris Volume Manager Administration Guide*.

CHAPTER 12

Preparing to Install From the Network (Topics)

This section provides instructions for setting up systems to install Solaris software from the local area network instead of DVD or CD media.

Chapter 13	Provides overview and planning information for installing Solaris software from an install server.
Chapter 14	Provides step-by-step instructions for copying Solaris software from DVD media to an install server.
Chapter 15	Provides step-by-step instructions for copying Solaris software from CD media to an install server.
Chapter 16	Describes commands to set up network installations.

CHAPTER 13

Preparing to Install From the Network (Overview)

This chapter provides an introduction on how to set up your local area network and systems to install the Solaris software from the network instead of from DVD or CD media.

For information on how to install a client over a wide area network, see Chapter 41.

Planning for a Network Installation Introduction

This section provides you with information you need before you can perform an installation from the network. Network installations enable you to install the Solaris software from a system, called an install server, that has access to the Solaris 9 disc images. You copy the contents of the Solaris 9 DVD or CD media to the install server's hard disk. Then, you can install the Solaris software from the network by using any of the Solaris installation methods.

Required Servers for Network Installation

To install the Solaris operating environment from the network, the systems to be installed require the following servers to be present on the network.

- **Install server –** A networked system that contains the Solaris 9 disc images from which you can install Solaris 9 software on other systems on the network. You create an install server by copying the images from the following media:
  - Solaris DVD
  - Solaris Software 1 of 2 CD and Solaris Software 2 of 2 CD

After you copy the image from the Solaris Software 1 of 2 and Solaris Software 2 of 2 CDs, you can also copy the images from the Solaris Installation CD and the Solaris Languages CD as necessary for your installation requirements.

You can enable a single install server to provide disc images for different Solaris releases and for multiple platforms by copying the images on to the install server's hard disk. For example, a single install server could contain the disc images for the SPARC platform and x86 platform.

For details about how to create an install server, refer to one of the following sections.

- "To Create a SPARC Install Server With SPARC or x86 DVD Media" on page 129
- "x86: To Create an x86 Install Server With SPARC or x86 DVD Media" on page 132
- "SPARC: To Create a SPARC Install Server With SPARC CD Media" on page 145
- "Creating a Cross-Platform Install Server for CD Media" on page 154

- **Boot server –** A server system that provides client systems on the same network subnet with the information that they need to boot in order to install the operating environment. A boot server and install server are typically the same system. However, if the system on which the Solaris 9 software is to be installed is located in a different subnet than the install server and you are not using DHCP, a boot server is required on that subnet.

  A single boot server can provide Solaris 9 boot software for multiple releases, including the Solaris 9 boot software for different platforms. For example, a SPARC boot server can provide the Solaris 8 and Solaris 9 boot software for SPARC based systems. The same SPARC boot server can also provide the Solaris 9 boot software for x86 based systems.

Note – When using DHCP, you do not need to create a separate boot server. For more information, see "Using DHCP to Provide Network Installation Parameters" on page 125.

For details about how to create a boot server, refer to one of the following sections:

- "Creating a Boot Server on a Subnet With a DVD Image" on page 135
- "Creating a Boot Server on a Subnet With a CD Image" on page 165

- **(Optional) Name server –** A system that manages a distributed network database, such as DNS, NIS, NIS+, or LDAP, that contains information about systems on the network.

  For details about how to create a name server, refer to *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Note – The install server and name server can be the same or different systems.

Figure 13–1 illustrates the servers that are typically used for network installation.

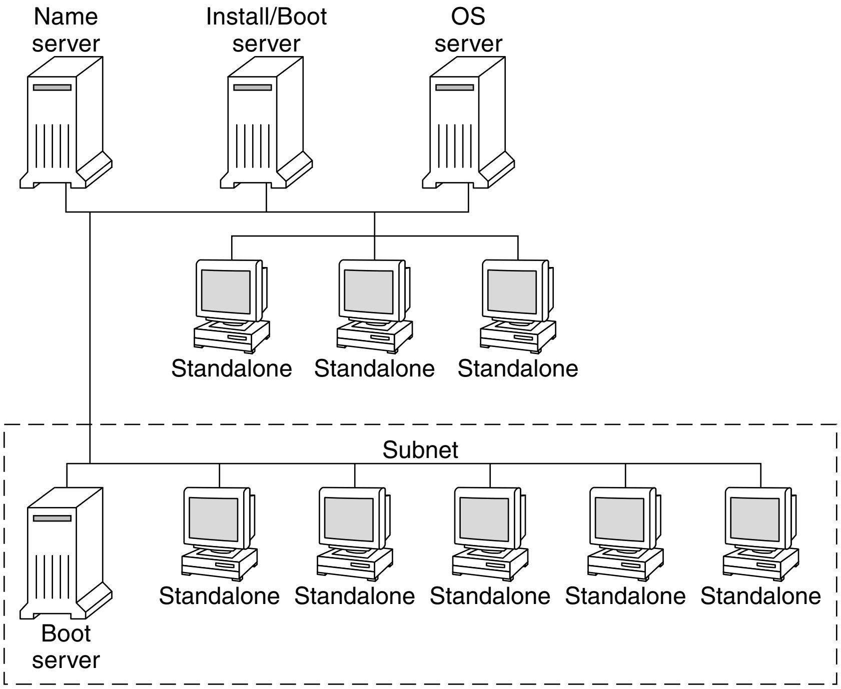


FIGURE 13–1 Network Installation Servers

Using DHCP to Provide Network Installation Parameters

Dynamic Host Configuration Protocol (DHCP) provides the network parameters that are necessary for installation. When using DHCP, you do not need to create a separate boot server. After you have created the install server, you add clients to the network with the `add_install_client` command and the `-d` option. The `-d` option enables you to set up client systems for Solaris installation from the network by using DHCP.

For information on DHCP options for installation parameters, see "Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85.

CHAPTER **14**

Preparing to Install From the Network With DVD Media (Tasks)

This chapter describes how to use DVD media to set up your network and systems to install the Solaris software from the network. Network installations enable you to install the Solaris software from a system that has access to the Solaris 9 disc images, called an install server, to other systems on the network. You copy the contents of the Solaris 9 DVD media to the install server's hard disk. Then, you can install the Solaris software from the network by using any of the Solaris installation methods. This chapter covers the following topics:

- "Task Map: Preparing to Install From the Network With DVD Media" on page 127
- "Creating an Install Server With DVD Media" on page 128
- "Creating a Boot Server on a Subnet With a DVD Image" on page 135
- "Adding Systems to Be Installed From the Network With a DVD Image" on page 137

Task Map: Preparing to Install From the Network With DVD Media

TABLE 14–1 Task Map: Setting Up an Install Server With DVD Media

Task	Description	For Instructions
Create an install server.	Use the `setup_install_server`(1M) command to copy the Solaris DVD to the install server's hard disk.	"Creating an Install Server With DVD Media" on page 128

TABLE 14–1 Task Map: Setting Up an Install Server With DVD Media *(Continued)*

Task	Description	For Instructions
(Optional) Create boot servers.	If you want to install systems from the network that are not on the same subnet as the install server, you must create a boot server on the subnet to boot the systems. Use the `setup_install_server` command with the `-b` option to set up a boot server. If you are using Dynamic Host Configuration Protocol (DHCP), a boot server is not necessary.	"Creating a Boot Server on a Subnet With a DVD Image" on page 135
Add systems to be installed from the network.	Use the `add_install_client` command to set up each system that you want to install from the network. Each system that you want to install needs to find the install server, the boot server if required, and configuration information on the network.	"Adding Systems to Be Installed From the Network With a DVD Image" on page 137

Creating an Install Server With DVD Media

The install server contains the installation image needed to install systems from the network. You must create an install server to install the Solaris software on a system from the network. You do not always need to set up a boot server.

- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a boot server.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet. However, install servers require more disk space.

Note – If you want use the Solaris DVD media to set up an install server on a system that is running the Solaris 2.6 or 7 operating environment, you must first apply one of the following patches.

- Solaris 2.6 *SPARC Platform Edition* operating environment - Patch ID 107618-03
- Solaris 2.6 *Intel Platform Edition* operating environment - Patch ID 107619-03
- Solaris 7 *SPARC Platform Edition* operating environment - Patch ID 107259-03
- Solaris 7 *Intel Platform Edition* operating environment - Patch ID 107260-03

▼ To Create a SPARC Install Server With SPARC or x86 DVD Media

---

Note – SPARC: You cannot use a system that is running a SunOS version that was released prior to the Solaris 2.3 release.

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

---

1. **On the SPARC system that is to become the install server, become superuser.**

   The system must include a DVD-ROM drive and be part of the site's network and name service. If you use a name service, the system must already be in a service, such as NIS, NIS+, DNS, or LDAP. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Insert the Solaris DVD in the SPARC system's drive.**

3. **Create a directory to contain the DVD image.**

   ```
   # mkdir -p install_dir_path
   ```

4. **Change to the `Tools` directory on the mounted disc.**

   - For SPARC DVD media, type:

     ```
     # cd /cdrom/cdrom0/s0/Solaris_9/Tools
     ```

   - For x86 DVD media, type:

     ```
     # cd /cdrom/cdrom0/Solaris_9/Tools
     ```

   In the previous examples, **`cdrom0`** is the path to the drive that contains the Solaris operating environment DVD media.

5. **Copy the DVD image in the drive to the install server's hard disk.**

   ```
   # ./setup_install_server install_dir_path
   ```

   install_dir_path Specifies the directory where the DVD image is to be copied

   ---

   Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

   ---

6. **Decide if you need to make the install server available for mounting.**
   - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. Proceed to Step 10.
   - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, complete the following steps.

7. **Verify that the path to the install server's image is shared appropriately.**

   ```
   # share | grep install_dir_path
   ```

   install_dir_path Specifies the path to the installation image where the DVD image was copied

   - If the path to the install server's directory is displayed and anon=0 is displayed in the options, proceed to Step 10.
   - If the path to the install server's directory is not displayed or you do not have anon=0 in the options, continue.

8. **Make the install server available to the boot server by adding this entry to the `/etc/dfs/dfstab` file.**

   ```
   share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
   ```

9. **Make sure that the install server's directory path is correctly shared.**

   ```
   # ps -ef | grep nfsd
   ```

   - If the `nfsd` daemon is not running, start it.

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

   - If the `nfsd` daemon is running, share the install server.

     ```
     # shareall
     ```

10. **Change directories to root (/).**

    ```
    # cd /
    ```

11. **Eject the Solaris DVD.**

12. **Decide if you want to patch the files that are located in the miniroot (/*install_dir_path*/`Solaris_9/Tools/Boot`) on the net install image that was created by `setup_install_server`. Patching a file might be necessary if a boot image has problems.**
    - If no, continue.
    - If yes, use the `patchadd -C` command to patch the files that are located in the miniroot.

Caution – Don't use the `patchadd -C` command unless you have read the Patch README instructions or have contacted your local Sun support office.

13. Decide if you need to create a boot server.

- If you are using DHCP or the install server is on the same subnet as the system to be installed, you do not need to create a boot server. Proceed to "Adding Systems to Be Installed From the Network With a DVD Image" on page 137.
- If you are *not* using DHCP and the install server and the client are on a different subnet, you must create a boot server. Proceed to "Creating a Boot Server on a Subnet With a DVD Image" on page 135.

EXAMPLE 14–1 SPARC: Creating a SPARC Install Server With a SPARC DVD

The following example illustrates how to create an install server by copying the Solaris DVD to the install server's `/export/home/dvdsparc` directory:

```
# mkdir -p /export/home/dvdsparc
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
# ./setup_install_server /export/home/dvdsparc
```

If you need a separate boot server, type these commands:

Add the following path to the `/etc/dfs/dfstab` file:

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdsparc
```

Check if the `nfsd` daemon is running. If the `nfsd` daemon is not running, start it and share it.

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

EXAMPLE 14–2 x86: Creating a SPARC Install Server With an x86 DVD

The following example illustrates how to create an install server by copying the Solaris DVD to the install server's `/export/home/dvdx86` directory:

```
# mkdir -p /export/home/dvdx86
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./setup_install_server /export/home/dvdx86
```

Add the following path to the `/etc/dfs/dfstab` file:

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdx86
```

EXAMPLE 14–2 x86: Creating a SPARC Install Server With an x86 DVD *(Continued)*

Check if the nfsd daemon is running. If the nfsd daemon is not running, start it and share it.

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

▼ x86: To Create an x86 Install Server With SPARC or x86 DVD Media

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

1. **On the x86 system that is to become the install server, become superuser.**

   The system must include a DVD-ROM drive and be part of the site's network and name service. If you use a name service, the system must also be in the NIS, NIS+, DNS, or LDAP name service. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Insert the Solaris DVD into the system's drive.**

3. **Create a directory to contain the boot image.**

   ```
   # mkdir -p install_dir_path
   ```

   install_dir_path Specifies the directory where the DVD image is to be copied

4. **Change to the `Tools` directory on the mounted disc:**

   - For x86 DVD media, type:

     ```
     # cd /cdrom/cdrom0/s2/Solaris_9/Tools
     ```

   - For SPARC DVD media, type:

     ```
     # cd /cdrom/cdrom0/Solaris_9/Tools
     ```

   In the previous examples, **`cdrom0`** is the path to the drive that contains the Solaris operating environment DVD media.

5. **Copy the disc in the drive to the install server's hard disk by using the `setup_install_server` command:**

```
# ./setup_install_server install_dir_path
```

install_dir_path Specifies the directory where the DVD image is to be copied

---

Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

---

6. **Decide if you need to make the install server available for mounting.**
   - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. Proceed to Step 10.
   - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, complete the following steps.

7. **Verify that the path to the install server's image is shared appropriately.**

   ```
   # share | grep install_dir_path
   ```

   install_dir_path Specifies the installation image where the DVD image was copied

   - If the path to the install server's directory is displayed and anon=0 is displayed in the options, proceed to Step 10.
   - If the path to the install server's directory is not displayed or you do not have anon=0 in the options, continue.

8. **Make the install server available to the boot server by adding this entry to the `/etc/dfs/dfstab` file.**

   ```
   share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
   ```

9. **Make sure that the install server's directory path is correctly shared.**

   ```
   # ps -ef | grep nfsd
   ```

   - If the `nfsd` daemon is not running, start it.

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

   - If the `nfsd` daemon is running, share the install server.

     ```
     # shareall
     ```

10. **Change directories to root (`/`).**

    ```
    # cd /
    ```

11. **Eject the Solaris DVD.**

12. **Decide if you want to patch the files that are located in the miniroot (`Solaris_9/Tools/Boot`) on the net install image that was created by `setup_install_server`.**

    - If no, continue.
    - If yes, use the `patchadd -C` command to patch the files that are located in the miniroot.

13. **Decide if you need to create a boot server.**

    - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. See "Adding Systems to Be Installed From the Network With a DVD Image" on page 137.
    - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must create a boot server. For detailed instructions on how to create a boot server, refer to "Creating a Boot Server on a Subnet With a DVD Image" on page 135.

EXAMPLE 14–3 x86: Creating an x86 Install Server With an x86 DVD

The following example illustrates how to create an x86 install server by copying the Solaris *x86 Platform Edition* DVD to the install server's `/export/home/dvdx86` directory:

```
# mkdir -p /export/home/dvdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/dvdx86
```

Add the following path to the `/etc/dfs/dfstab` file:

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdx86
```

Check if the `nfsd` daemon is running. If the `nfsd` daemon is not running, start it and share it.

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

EXAMPLE 14–4 Creating an x86 Install Server With a SPARC DVD

The following example illustrates how to create an x86 install server by copying the Solaris *SPARC Platform Edition* DVD to the install server's `/export/home/dvdsparc` directory:

```
# mkdir -p /export/home/dvdscparc
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./setup_install_server /export/home/dvdsparc
```

Add the following path to the `/etc/dfs/dfstab` file:

EXAMPLE 14–4 Creating an x86 Install Server With a SPARC DVD *(Continued)*

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdsparc
```

Check if the nfsd daemon is running. If the nfsd daemon is not running, start it and share it.

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

Creating a Boot Server on a Subnet With a DVD Image

You must create an install server to install the Solaris software on a system from the network. You do not always need to set up a boot server. A boot server contains enough of the boot software to boot systems from the network, and then the install server completes the installation of the Solaris software.

- If you are using DHCP to set installation parameters or your install server or client is on the same subnet as the install server, you do not need a boot server. Proceed to "Adding Systems to Be Installed From the Network With a DVD Image" on page 137.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.

▼ To Create a Boot Server on a Subnet With a DVD Image

1. **On the system you intend to make the boot server for the subnet, log in and become superuser.**

   The system must have access to a remote Solaris 9 disc image, which is normally the install server. If you use a name service, the system should also be in a name service. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Mount the Solaris DVD from the install server.**

```
# mount -F nfs -o ro server_name:path /mnt
```

server_name:*path* Is the install server name and absolute path to the disc image

3. **Create a directory for the boot image.**

```
#  mkdir -p boot_dir_path
```

boot_dir_path Specifies the directory where the boot software is to be copied

4. **Change to the `Tools` directory on the Solaris DVD image.**

```
# cd /mnt/Solaris_9/Tools
```

5. **Copy the boot software to the boot server.**

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

`-b` Specifies to set up the system as a boot server

boot_dir_path Specifies the directory where the boot software is to be copied

---

Note – The `setup_install_server` command indicates whether you have enough disk space available for the images. To determine available disk space, use the `df -kl` command.

---

6. **Change directories to root (/).**

```
# cd /
```

7. **Unmount the installation image.**

```
# umount /mnt
```

You are now ready to set up systems to be installed from the network. See "Adding Systems to Be Installed From the Network With a DVD Image" on page 137.

EXAMPLE 14–5 Creating a Boot Server on a Subnet (DVD)

The following example illustrates how to create a boot server on a subnet. These commands copy the boot software from the Solaris DVD image to `/export/home/dvdsparc` on the local disk of a boot server named `crystal`.

```
# mount -F nfs -o ro crystal:/export/home/dvdsparc /mnt
# mkdir -p  /export/home/dvdsparc
# cd /mnt/Solaris_9/Tools
# ./setup_install_server -b /export/home/dvdsparc
# cd /
# umount /mnt
```

Adding Systems to Be Installed From the Network With a DVD Image

After you create an install server and, if necessary, a boot server, you must set up each system that you want to install from the network. Each system that you want to install needs to find the following:

- Install server
- Boot server if it is required
- sysidcfg file if you use a sysidcfg file to preconfigure system information
- Name server if you use a name service to preconfigure system information
- The profile in the JumpStart directory on the profile server if you are using the custom JumpStart installation method

Use the following add_install_client procedure for setting up install servers and clients. Also, see the example procedures for the following:

- If you are using DHCP to set installation parameters, see Example 14–6.
- If your install server and client are on the same subnet, see Example 14–7.
- If your install server and your client are not on the same subnet and you are not using DHCP, see Example 14–8.
- If you are using DHCP to set installation parameters for x86 clients, see Example 14–9.
- If you want to use a specific serial port to display output during the installation of an x86-based system, see Example 14–10.
- If you want to set up an x86 client to use a specific network interface during the installation, see Example 14–11.

For more options to use with this command, see the man page, add_install_client(1M).

▼ To Add Systems to Be Installed From the Network With add_install_client (DVD)

If you have a boot server, make sure you have shared the install server installation image and started the appropriate daemons. See "To Create a SPARC Install Server With SPARC or x86 DVD Media" Step 6.

1. **On the install server or boot server, become superuser.**

2. **If you use the NIS, NIS+, DNS, or LDAP name service, verify that the following information about the system to be installed has been added to the name service.**

   - Host name
   - IP address
   - Ethernet address

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

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

   ```
   # cd /install_dir_path/Solaris_9/Tools
   ```

   install_dir_path Specifies the path to the `Tools` directory

4. **Set up the client system so it can be installed from the network.**

   ```
   # ./add_install_client [-d] [-s install_server:install_dir_path] \
   [-c jumpstart_server:jumpstart_dir_path]  [-p sysid_server:path] \
   [-t boot_image_path] [-b "boot-property=value"] \
   [-e ethernet_address] client_name platform_group
   ```

   `-d`

   Specifies that the client is to use DHCP to obtain the network install parameters. If you use the `-d` only, the `add_install_client` command sets up the installation information for client systems of the same class, for example, all SPARC client machines. To set up the installation information for a specific client, use the `-d` with the `-e` option.

   For x86 clients, use this option to boot the systems from the network by using PXE network boot.

   For more information about class-specific installations by using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86.

   `-s` *install_server:install_dir_path*

   Specifies the name and path to the install server.

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

   `-c` *jumpstart_server* : *jumpstart_dir_path*

   Specifies a JumpStart directory for custom JumpStart installations.

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

`-p` *sysid_server*`:`*path*

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

`-t` *boot_image_path*

Specifies the path to an alternate boot image if you want to use a boot image other than the one in the `Tools` directory on the Solaris 9 net installation image, CD, or DVD.

`-b` "*boot-property=value*"

x86 based systems only: Enables you to set the value of a boot property variable that you want to use to boot the client from the network. The `-b` option must be used with the `-e` option.

See the `eeprom`(1M) man page for descriptions of boot properties.

`-e` *ethernet_address*

Specifies the ethernet address of the client that you want to install. This option enables you to set up the installation information to use for a specific client.

For more information about client-specific installations by using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86.

client_name

Is the name of the system to be installed from the network. This name is *not* the host name of the install server.

platform_group

Is the platform group of the system to be installed. For more information, see Appendix C.

EXAMPLE 14–6 SPARC: Adding a SPARC Install Client on a SPARC Install Server When Using DHCP (DVD)

The following example illustrates how to add an install client when you are using DHCP to set installation parameters on the network. The install client is named `basil`, which is an Ultra™ 5 system. The file system `/export/home/dvdsparc/Solaris_9/Tools` contains the `add_install_client` command.

For more information on how to use DHCP to set installation parameters for network installations, see "Supporting Solaris Network Installation with the DHCP Service (Task Map)" in *System Administration Guide: IP Services*.

```
sparc_install_server# cd /export/home/dvdsparc/Solaris_9/Tools
sparc_install_server# ./add_install_client -d basil sun4u
```

EXAMPLE 14–7 Adding an Install Client That Is On the Same Subnet As Its Server (DVD)

The following example illustrates how to add an install client that is on the same subnet as the install server. The install client is named `basil`, which is an Ultra 5 system. The file system `/export/home/dvdsparc/` contains the `add_install_client` command.

```
install_server# cd /export/home/dvdsparc/Solaris_9/Tools
install_server# ./add_install_client basil sun4u
```

EXAMPLE 14–8 Adding an Install Client to a Boot Server (DVD)

The following example illustrates how to add an install client to a boot server. The install client is named `rose`, which is an Ultra 5 system. Run the command on the boot server. The `-s` option is used to specify an install server that is named `rosemary`, which contains a Solaris *SPARC Platform Edition* DVD image in `/export/home/dvdsparc`.

```
boot_server# cd /export/home/dvdsparc/Solaris_9/Tools
boot_server# ./add_install_client -s rosemary:/export/home/dvdsparc rose sun4u
```

EXAMPLE 14–9 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP (DVD)

The following example illustrates how to add an x86 install client to an install server when you are using DHCP to set installation parameters on the network. The `-d` option is used to specify that clients are to use the DHCP protocol for configuration. If you plan to use PXE network boot, you must use the DHCP protocol. The DHCP class name SUNW.i86pc indicates that this command applies to all Solaris x86 network boot clients, not just a single client. The `-s` option is used to specify that the clients are to be installed from the install server that is named `rosemary`. This server contains a Solaris *x86 Platform Edition* DVD image in `/export/home/dvdx86`.

EXAMPLE 14–9 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP (DVD) *(Continued)*

For more information on how to use DHCP to set installation parameters for network installations, see "Supporting Solaris Network Installation with the DHCP Service (Task Map)" in *System Administration Guide: IP Services*.

```
x86_install_server# cd /export/boot/dvdx86/Solaris_9/Tools
x86_install_server# ./add_install_client -d -s rosemary:/export/home/dvdx86 \
SUNW.i86pc i86pc
```

EXAMPLE 14–10 x86: Specifying a Serial Console to Use During a Network Installation (DVD)

The following example illustrates how to add an x86 install client to an install server and specify a serial console to use during the installation. This example sets up the install client in the following manner.

- The `-d` option indicates that the client is set up to use DHCP to set installation parameters.
- The `-e` option indicates that this installation will occur only on the client with the ethernet address 00:07:e9:04:4a:bf.
- The first and second uses of the `-b` option instruct the installation program to use the serial port `ttya` as an input and an output device.

```
install server# cd /export/boot/dvdx86/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" i86pc
```

For a complete description of the boot property variables and values you can use with the `-b` option, see the `eeprom`(1M) man page.

EXAMPLE 14–11 x86: Specifying a Boot Device to Use During a Network Installation (DVD)

The following example illustrates how to add an x86 install client to an install server and specify a boot device to use during the installation. If you specify the boot device when you set up the install client, you are not prompted for this information by the Device Configuration Assistant during the installation.

This example sets up the install client in the following manner.

- The `-d` option indicates that the client is set up to use DHCP to set installation parameters.
- The `-e` option indicates that this installation will occur only on the client with the ethernet address 00:07:e9:04:4a:bf.
- The first and second uses of the `-b` option instruct the installation program to use the serial port `ttya` as an input and an output device.
- The third use of the `-b` option instructs the installation program to use a specific boot device during the installation.

Note – The value of the boot device path varies based on your hardware.

- The i86pc platform name indicates that the client is an x86-based system.

```
install server# cd /export/boot/dvdx86/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

For a complete description of the boot property variables and values you can use with the `-b` option, see the `eeprom`(1M) man page.

CHAPTER 15

Preparing to Install From the Network With CD Media (Tasks)

This chapter describes how to use CD media to set up your network and systems to install the Solaris software from the network. Network installations enable you to install the Solaris software from a system that has access to the Solaris 9 disc images, called an install server, to other systems on the network. You copy the contents of the CD media to the install server's hard disk. Then, you can install the Solaris software from the network by using any of the Solaris installation methods. This chapter covers the following topics:

- "Task Map: Preparing to Install From the Network With CD Media" on page 144
- "Creating a SPARC Install Server With CD Media" on page 144
- "Creating an x86 Install Server With CD Media" on page 149
- "Creating a Cross-Platform Install Server for CD Media" on page 154
- "Creating a Boot Server on a Subnet With a CD Image" on page 165
- "Adding Systems to Be Installed From the Network With a CD Image" on page 167

Task Map: Preparing to Install From the Network With CD Media

TABLE 15–1 Task Map: Setting Up an Install Server With CD Media

Task	Description	For Instructions
Create an install server.	Use the `setup_install_server`(1M) command to copy the Solaris Software 1 of 2 CD to the install server's hard disk. Use the `add_to_install_server`(1M) command to copy the Solaris Software 2 of 2 CD and the Solaris Languages CD to the install server's hard disk. Use the `modify_install_server`(1M) command to add the Solaris Web Start user interface software to the net installation image.	■ "Creating a SPARC Install Server With CD Media" on page 144 ■ "x86: To Create an x86 Install Server With x86 CD Media" on page 149 ■ "Creating a Cross-Platform Install Server for CD Media" on page 154
(Optional) Create boot servers.	If you want to install systems from the network that are not on the same subnet as the install server, you must create a boot server on the subnet to boot the systems. If you are using Dynamic Host Configuration Protocol (DHCP), a boot server is not necessary.	"Creating a Boot Server on a Subnet With a CD Image" on page 165
Add systems to be installed from the network.	Use the `add_install_client` command to set up each system that you want to install from the network. Each system that you want to install needs to find the install server, the boot server if required, and configuration information on the network.	"Adding Systems to Be Installed From the Network With a CD Image" on page 167

Creating a SPARC Install Server With CD Media

The install server contains the installation image needed to install systems from the network. You must create an install server to install the Solaris software on a system from the network. You do not always need to set up a separate boot server.

- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a separate boot server.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.

▼ SPARC: To Create a SPARC Install Server With SPARC CD Media

This procedure creates a SPARC install server with SPARC CD media.

If you want to create an install server by using media of a platform different from the install server, for example, a SPARC system with x86 CD media, see "Creating a Cross-Platform Install Server for CD Media" on page 154.

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

---

1. **On the system that is to become the install server, become superuser.**

   The system must include a CD-ROM drive and be part of the site's network and name service. If you use a name service, the system must already be in a name service, such as NIS, NIS+, DNS, or LDAP. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Insert the Solaris Software 1 of 2 CD in the system's drive.**

3. **Create a directory for the CD image.**

   ```
   # mkdir -p install_dir_path
   ```

   install_dir_path Specifies the directory where the CD image is to be copied

4. **Change to the `Tools` directory on the mounted disc.**

   ```
   # cd /cdrom/cdrom0/s0/Solaris_9/Tools
   ```

   In the previous example, **`cdrom0`** is the path to the drive that contains the Solaris operating environment CD media.

5. **Copy the image in the drive to the install server's hard disk.**

   ```
   # ./setup_install_server install_dir_path
   ```

   install_dir_path Specifies the directory where the CD image is to be copied

---

Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

---

6. **Decide if you need to make the install server available for mounting.**
   - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. Proceed to Step 10.
   - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, complete the following steps.

7. **Verify that the path to the install server's installation image is shared appropriately.**

   ```
   # share | grep install_dir_path
   ```

   install_dir_path Specifies the path to the boot-server installation image

   - If the path to the install server's directory is displayed and anon=0 is displayed in the options, proceed to Step 10.
   - If the path to the install server's directory is not displayed or you do not have anon=0 in the options, continue.

8. **Make the install server available to the boot server by adding this entry to the `/etc/dfs/dfstab` file.**

   ```
   share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
   ```

   install_dir_path Specifies the path to the boot server installation image

9. **Make sure that the install server's directory path is correctly shared.**

   ```
   # ps -ef | grep nfsd
   ```

   - If the `nfsd` daemon is not running, start it.

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

   - If the `nfsd` daemon is running, share the install server.

     ```
     # shareall
     ```

10. **Change directories to root (/).**

    ```
    # cd /
    ```

11. **Eject the Solaris Software 1 of 2 CD.**

12. **Insert the Solaris Software 2 of 2 CD in the system's CD-ROM drive.**

13. **Change to the `Tools` directory on the mounted CD.**

```
# cd /cdrom/cdrom0/Solaris_9/Tools
```

14. **Copy the CD in the CD-ROM drive to the install server's hard disk.**

```
# ./add_to_install_server install_dir_path
```

install_dir_path Specifies the directory where the CD image is to be copied

15. **Change directories to root (/).**

```
# cd /
```

16. **Eject the Solaris Software 2 of 2 CD.**

17. **Insert the Solaris Languages CD in the system's CD-ROM drive.**

18. **Change to the `Tools` directory on the mounted CD.**

```
# cd /cdrom/cdrom0/Tools
```

19. **Copy the CD in the CD-ROM drive to the install server's hard disk.**

```
# ./add_to_install_server install_dir_path
```

install_dir_path Specifies the directory where the CD image is to be copied

20. **Change directories to root (/).**

```
# cd /
```

21. **Decide if you want to enable users to use the Solaris Web Start installation method to boot a system and install the Solaris 9 software from a network.**
    - If no, eject the Solaris Languages CD and proceed to Step 25.
    - If yes, eject the Solaris Languages CD and continue.

22. **Insert the Solaris Installation CD into the system's CD-ROM drive.**

23. **Change to the directory that contains `modify_install_server` on the mounted CD:**

```
# cd /cdrom/cdrom0/s0
```

24. **Copy the Solaris Web Start interface software to the install server.**

```
# ./modify_install_server -p install_dir_path installer_miniroot_path
```

`-p` Preserves the existing image's miniroot in *install_dir_path*`/Solaris_9/Tools/Boot.orig`

install_dir_path Specifies the directory where the Solaris Web Start interface is to be copied

installer_miniroot_path Specifies the directory on the CD from which the Solaris Web Start interface is to be copied

25. **Decide if you want to patch the files that are located in the miniroot (/*install_dir_path*/Solaris_9/Tools/Boot) on the net install image that was created by setup_install_server. Patching a file might be necessary if a boot image has problems.**

    - If no, continue.
    - If yes, use the `patchadd -C` command to patch the files that are located in the miniroot.

    Caution – Don't use the `patchadd -C` command unless you have read the `Patch README` instructions or have contacted your local Sun support office.

26. **Decide if you need to create a boot server.**

    - If you are using DHCP or the install server is on the same subnet as the system to be installed, you do not need to create a boot server. Proceed to "Adding Systems to Be Installed From the Network With a CD Image" on page 167.
    - If you are *not* using DHCP and the install server and the client are on a different subnet, you must create a boot server. Proceed to "Creating a Boot Server on a Subnet With a CD Image" on page 165.

EXAMPLE 15–1 SPARC: Creating a SPARC Install Server With SPARC CD Media

The following example illustrates how to create an install server by copying the following CDs to the install server's `/export/home/cdsparc` directory:

- Solaris Software 1 of 2 *SPARC Platform Edition* CD
- Solaris Software 2 of 2 *SPARC Platform Edition* CD
- Solaris *SPARC Platform Edition* Languages CD

```
# mkdir -p /export/home/cdsparc
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
# ./setup_install_server /export/home/cdsparc
```

- If you have a separate boot server, add these steps.

  1. Add the following path to the `/etc/dfs/dfstab` file.

     ```
     share -F nfs -o ro,anon=0 -d "install server directory" \
     /export/home/cdsparc
     ```

  2. Check if the `nfsd` daemon is running. If the `nfsd` daemon is not running, start it and share it.

     ```
     # ps -ef | grep nfsd
     # /etc/init.d/nfs.server start
     # shareall
     ```

  3. Continue with the following steps.

EXAMPLE 15–1 SPARC: Creating a SPARC Install Server With SPARC CD Media *(Continued)*

- If you do not need a boot server or have completed the steps for a separate boot server, continue.

```
# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./add_to_install_server /export/home/cdsparc
# cd /
# cd /cdrom/cdrom0/Tools
# ./add_to_install_server /export/home/cdsparc
# cd /
# cd /cdrom/cdrom0/s0
# ./modify_install_server -p /export/home/cdsparc /cdrom/cdrom0/s0
```

In this example, each CD is inserted and automatically mounted before each of the commands. After each command, the CD is removed.

Creating an x86 Install Server With CD Media

The install server contains the installation image needed to install systems from the network. You must create an install server to install the Solaris software on a system from the network. You do not always need to set up a separate boot server.

- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a separate boot server.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.

x86: To Create an x86 Install Server With x86 CD Media

This procedure creates an x86 install server with x86 CD media.

If you want to create an install server by using media of a platform different from the install server, for example, an x86 system with SPARC CD media, see “Creating a Cross-Platform Install Server for CD Media” on page 154.

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

1. **On the system that is to become the install server, become superuser.**

   The system must include a CD-ROM drive and be part of the site's network and name service. If you use a name service, the system must already be in a name service, such as NIS, NIS+, DNS, or LDAP. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Insert the Solaris Software 1 of 2 CD in the system's drive.**

3. **Create a directory for the CD image.**

   ```
   # mkdir -p install_dir_path
   ```

   install_dir_path Specifies the directory where the CD image is to be copied

4. **Change to the `Tools` directory on the mounted disc.**

   ```
   # cd /cdrom/cdrom0/s2/Solaris_9/Tools
   ```

   In the previous example, **`cdrom0`** is the path to the drive that contains the Solaris operating environment CD media.

5. **Copy the image in the drive to the install server's hard disk.**

   ```
   # ./setup_install_server install_dir_path
   ```

   install_dir_path Specifies the directory where the CD image is to be copied

   Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

6. **Decide if you need to make the install server available for mounting.**

   - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. Proceed to Step 10.
   - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, complete the following steps.

7. **Verify that the path to the install server's installation image is shared appropriately.**

   ```
   # share | grep install_dir_path
   ```

install_dir_path Specifies the path to the boot-server installation image

- If the path to the install server's directory is displayed and anon=0 is displayed in the options, proceed to Step 10.
- If the path to the install server's directory is not displayed or you do not have anon=0 in the options, continue.

8. **Make the install server available to the boot server by adding this entry to the `/etc/dfs/dfstab` file.**

   ```
   share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
   ```

   install_dir_path Specifies the path to the boot-server installation image

9. **Make sure that the install server's directory path is correctly shared.**

   ```
   # ps -ef | grep nfsd
   ```

   - If the `nfsd` daemon is not running, start it.

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

   - If the `nfsd` daemon is running, share the install server.

     ```
     # shareall
     ```

10. **Change directories to root (/).**

    ```
    # cd /
    ```

11. **Eject the Solaris Software 1 of 2 CD.**

12. **Insert the Solaris Software 2 of 2 CD in the system's CD-ROM drive.**

13. **Change to the `Tools` directory on the mounted CD:**

    ```
    # cd /cdrom/cdrom0/Solaris_9/Tools
    ```

14. **Copy the CD in the CD-ROM drive to the install server's hard disk.**

    ```
    # ./add_to_install_server install_dir_path
    ```

    install_dir_path Specifies the directory where the CD image is to be copied

15. **Change directories to root (/).**

    ```
    # cd /
    ```

16. **Eject the Solaris Software 2 of 2 CD.**

17. **Insert the Solaris Languages CD in the system's CD-ROM drive.**

18. **Change to the `Tools` directory on the mounted CD:**

    ```
    # cd /cdrom/cdrom0/Tools
    ```

19. **Copy the CD in the CD-ROM drive to the install server's hard disk.**

    ```
    # ./add_to_install_server install_dir_path
    ```

    install_dir_path Specifies the directory where the CD image is to be copied

20. **Change directories to root (/).**

    ```
    # cd /
    ```

21. **Decide if you want to enable users to use the Solaris Web Start installation method to boot a system and install the Solaris 9 software from a network.**

    - If no, eject the Solaris Languages CD and proceed to Step 25.
    - If yes, eject the Solaris Languages CD and continue.

22. **Insert the Solaris Installation CD into the system's CD-ROM drive.**

23. **Change to the directory that contains `modify_install_server` on the mounted CD:**

    ```
    # cd /cdrom/cdrom0/s2
    ```

24. **Copy the Solaris Web Start interface software to the install server.**

    ```
    # ./modify_install_server -p install_dir_path installer_miniroot_path
    ```

    `-p` Preserves the existing image's miniroot in *install_dir_path*`/Solaris_9/Tools/Boot.orig`

    install_dir_path Specifies the directory where the Solaris Web Start interface is to be copied

    installer_miniroot_path Specifies the directory on the CD from which the Solaris Web Start interface is to be copied

25. **Decide if you want to patch the files that are located in the miniroot (/*install_dir_path*/`Solaris_9/Tools/Boot`) on the net install image that was created by `setup_install_server`. Patching a file might be necessary if a boot image has problems.**

    - If no, continue.
    - If yes, use the `patchadd -C` command to patch the files that are located in the miniroot.

    Caution – Don't use the `patchadd -C` command unless you have read the `Patch README` instructions or have contacted your local Sun support office.

    ---

26. **Decide if you need to create a boot server.**

- If you are using DHCP or the install server is on the same subnet as the system to be installed, you do not need to create a boot server. Proceed to “Adding Systems to Be Installed From the Network With a CD Image” on page 167.
- If you are *not* using DHCP and the install server and the client are on a different subnet, you must create a boot server. Proceed to “Creating a Boot Server on a Subnet With a CD Image” on page 165.

EXAMPLE 15–2 x86: Creating an x86 Install Server With x86 CD Media

The following example illustrates how to create an install server by copying the following CDs to the install server’s `/export/home/cdx86` directory:

- Solaris Software 1 of 2 *x86 Platform Edition* CD
- Solaris Software 2 of 2 *x86 Platform Edition* CD
- Solaris *x86 Platform Edition* Languages CD

```
# mkdir -p /export/home/cdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/cdx86
```

- If you have a separate boot server, add these steps.
  1. Add the following path to the `/etc/dfs/dfstab` file:

     ```
     share -F nfs -o ro,anon=0 -d "install server directory" \
     /export/home/cdx86
     ```

  2. Check if the `nfsd` daemon is running. If the `nfsd` daemon is not running, start it and share it.

     ```
     # ps -ef | grep nfsd
     # /etc/init.d/nfs.server start
     # shareall
     ```

  3. Continue with the following steps.
- If you do not need a boot server or have completed the steps for a separate boot server, continue with the following steps.

```
# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/s2
# ./modify_install_server -p /export/home/cdx86 /cdrom/cdrom0/s2
```

In this example, each CD is inserted and automatically mounted before each of the commands. After each command, the CD is removed.

Creating a Cross-Platform Install Server for CD Media

If you need to use a CD of a platform different from the install server, you cannot read the CD in the install server. You need a remote system to read the CD. For example, if you are setting up a SPARC install server and need to use x86 CD media, you need a remote x86 system to read the CDs.

▼ To Create an x86 Install Server on a SPARC System With x86 CD Media

Use this procedure to create an x86 install server on a SPARC system with x86 CD media. You need the following:

- A SPARC system
- An x86 system with a CD-ROM drive
- A set of CDs for the remote x86 system
  - Solaris Software 1 of 2 *x86 Platform Edition* CD
  - Solaris Software 2 of 2 *x86 Platform Edition* CD
  - Solaris *x86 Platform Edition* Languages CD
  - Solaris *x86 Platform Edition* Installation CD

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

In this procedure, *SPARC-system* is the SPARC system that is to be the install server and *remote-x86-system* is the remote x86 system to be used with the x86 CD media.

1. **On the remote x86 system, become superuser.**

   The system must include a CD-ROM drive and be part of the site's network and name service. If you use a name service, the system must also be in the NIS, NIS+, DNS, or LDAP name service. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **On the remote x86 system, insert the Solaris Software 1 of 2** *x86 Platform Edition* **CD into the system's drive.**

3. **On the remote x86 system, add the following entries to the `/etc/dfs/dfstab` file.**

   ```
   share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
   share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
   ```

4. **On the remote x86 system, start the NFS daemon.**

   ```
   remote-x86-system# /etc/init.d/nfs.server start
   ```

5. **On the remote x86 system, verify that the CD is available to other systems by using the share command.**

   ```
   remote-x86-system# share
   -    /cdrom/sol_9_x86/s0 ro,anon=0 " "
   -    /cdrom/sol_9_x86/s2 ro,anon=0 " "
   ```

   In the previous sample output, `sol_9_x86` refers to the Solaris 9 (*x86 Platform Edition*) operating environment. This text string varies for each version of the Solaris operating environment.

6. **On the SPARC system that is to be the x86 install server, become superuser.**

7. **On the SPARC system, access the x86 CD by creating two directories for the appropriate mount points, one for the miniroot and one for the product.**

   ```
   SPARC-system# mkdir directory_name_s0
   ```

   ```
   SPARC-system# mkdir  directory_name_s2
   ```

   directory_name_s0 Is the name of the directory to contain the miniroot from slice 0

   directory_name_s2 Is the name of the directory to contain the product from slice 2

8. **Verify that the CD is properly exported on the remote x86 system.**

   ```
   SPARC-system# showmount -e remote-x86-system
   export list for remote-x86-system:
   /cdrom/sol_9_x86/s0 (everyone)
   /cdrom/sol_9_x86/s2 (everyone)
   ```

9. **On the SPARC system, mount the remote x86 CD image.**

```
SPARC-system# mount remote_x86_system_name:/cdrom/sol_9_x86/s0 directory_name_s0
```

```
SPARC-system# mount remote_x86_system_name:/cdrom/sol_9_x86/s2 directory_name_s2
```

10. **On the SPARC system, change to the `Tools` directory on the mounted disc:**

    ```
    SPARC-system# cd /directory_name_s2/Solaris_9/Tools
    ```

11. **On the SPARC system, copy the disc in the drive to the install server's hard disk in the directory you've created by using the `setup_install_server` command:**

```
SPARC-system# ./setup_install_server -t directory_name_s0 install_dir_path
```

`-t` Specifies the path to a boot image if you want to use a boot image other than the one in the `Tools` directory on the Solaris Software 2 of 2 CD.

directory_name_s0 Is the name of the directory that contains the miniroot from slice 0.

install_dir_path Specifies the directory where the disc image is to be copied. The directory must be empty.

Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

12. **On the SPARC system, change to the top directory.**

    ```
    SPARC-system# cd /
    ```

13. **On the SPARC system, unmount both directories.**

    ```
    SPARC-system# unmount directory_name_s0
    ```

    ```
    SPARC-system# unmount directory_name_s2
    ```

14. **On the x86 system, unshare both CD-ROM slices.**

    ```
    remote x86 system# unshare /cdrom/sol_9_x86/s0
    ```

    ```
    remote x86 system# unshare /cdrom/sol_9_x86/s2
    ```

15. **On the x86 system, eject the Solaris Software 1 of 2** *x86 Platform Edition* **CD.**

16. **Insert the Solaris Software 2 of 2** *x86 Platform Edition* **CD into the SPARC system's CD-ROM drive.**

17. **On the SPARC system, change to the `Tools` directory on the mounted CD:**

    ```
    SPARC-system# cd /cdrom/cdrom0/Solaris_9/Tools
    ```

18. **On the SPARC system, copy the CD to the install server's hard disk:**

    ```
    SPARC-system# ./add_to_install_server install_dir_path
    ```

    install_dir_path Specifies the directory where the CD image is to be copied

19. **Eject the Solaris Software 2 of 2** *x86 Platform Edition* **CD.**

20. **On the SPARC system, insert the Solaris** *x86 Platform Edition* **Languages CD into the SPARC system's CD-ROM drive and mount the CD.**

21. **On the SPARC system, change to the `Tools` directory on the mounted CD:**

    ```
    SPARC-system# cd /cdrom/cdrom0/Tools
    ```

22. **On the SPARC system, copy the CD to the install server's hard disk:**

    ```
    SPARC-system# ./add_to_install_server install_dir_path
    ```

    install_dir_path Specifies the directory where the CD image is to be copied

23. **Decide if you want to enable users to use the Solaris Web Start installation method to boot a system and install the Solaris 9 software from a network.**

    - If no, eject the Solaris *x86 Platform Edition* Languages CD and proceed to Step 31.
    - If yes, eject the Solaris *x86 Platform Edition* Languages CD and continue.

24. **On the remote x86 system, insert the Solaris *x86 Platform Edition* Installation CD into the x86 system's CD-ROM drive.**

25. **On the remote x86 system, add the following entry to the `/etc/dfs/dfstab` file.**

    ```
    share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
    share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
    ```

26. **On the remote x86 system, start the NFS daemon.**

    ```
    remote-x86-system# /etc/init.d/nfs.server start
    ```

27. **On the remote x86 system, verify that the CD is available to other systems:**

    ```
    remote-x86-system# share
    -   /cdrom/multi_sol_9_x86/s0 ro,anon=0 " "
    -   /cdrom/multi_sol_9_x86/s2 ro,anon=0 " "
    ```

28. **On the SPARC system, mount the slices on the CD.**

```
SPARC-system# mount remote_x86_system_name:/cdrom/multi_sol_9_x86/s0 directory_name_s0
SPARC-system# mount remote_x86_system_name:/cdrom/multi_sol_9_x86/s2  directory_name_s2
```

29. **On the SPARC system, change to the directory that contains `modify_install_server` on the mounted CD:**

    ```
    SPARC-system# cd directory_name_s2
    ```

30. **Copy the Solaris Web Start interface software to the install server.**

    ```
    SPARC-system# ./modify_install_server -p install_dir_path directory_name_s0
    ```

    `-p` Preserves the existing image's miniroot in *install_dir_path*`/Solaris_9/Tools/Boot.orig`

    install_dir_path Specifies the directory where the Solaris Web Start interface is to be copied *directory_name_s0*   Name of the directory to contain the miniroot from slice 0

31. **Decide if you want to patch the files that are located in the miniroot (`Solaris_9/Tools/Boot`) on the net installation image that was created by `setup_install_server`.**
    - If no, proceed to the next step.
    - If yes, use the `patchadd -C` command to patch the files that are located in the miniroot.

> **Caution –** Don't use the `patchadd -C` command unless you have read the `Patch README` instructions or have contacted your local Sun support office.

32. **Decide if you need to create a boot server.**
    - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. See "Adding Systems to Be Installed From the Network With a CD Image" on page 167.
    - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must create a boot server. For detailed instructions on how to create a boot server, refer to "To Create a Boot Server on a Subnet With a CD Image" on page 165.

EXAMPLE 15–3 Creating an x86 Install Server on a SPARC System With x86 CD Media

The following example illustrates how to create an x86 install server on a SPARC system that is named `rosemary`. The following x86 CDs are copied from a remote x86 system that is named `tadpole` to the SPARC install server's `/export/home/cdx86` directory.

- Solaris Software 1 of 2 *x86 Platform Edition* CD
- Solaris Software 2 of 2 *x86 Platform Edition* CD
- Solaris *x86 Platform Edition* Languages CD
- Solaris Installation Multilingual *x86 Platform Edition* CD

On the remote x86 system:

```
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
tadpole (remote-x86-system)# /etc/init.d/nfs.server start
```

On the SPARC system:

```
rosemary (SPARC-system)# mkdir /x86S0
rosemary (SPARC-system)# mkdir /x86S2
rosemary (SPARC-system)# mount tadpole:/cdrom/sol_9_x86/s0 /x86S0
rosemary (SPARC-system)# mount tadpole:/cdrom/sol_9_x86/s0 /x86S2
rosemary (SPARC-system)# cd /x86S2/Solaris_9/Tools
```

EXAMPLE 15–3 Creating an x86 Install Server on a SPARC System With x86 CD Media *(Continued)*

```
rosemary (SPARC-system)# ./setup_install_server -t /x86S0 /export/home/cdx86
rosemary (SPARC-system)# cd /
rosemary (SPARC-system)# unmount /x86S0
rosemary (SPARC-system)# unmount /x86S2

tadpole (remote-x86-system) unshare /cdrom/cdrom0/s0
tadpole (remote-x86-system) unshare  /cdrom/cdrom0/s2

rosemary (SPARC-system)# cd /cdrom/cdrom0/Solaris_9/Tools
rosemary (SPARC-system)# ./add_to_install_server /export/home/cdx86
rosemary (SPARC-system)# cd /cdrom/cdrom0/Tools
rosemary (SPARC-system)# ./add_to_install_server /export/home/cdx86
```

On the remote x86 system:

```
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
tadpole (remote-x86-system)# /etc/init.d/nfs.server start
```

On the SPARC system:

```
rosemary (SPARC-system)# mount remote_x86_system_name:/cdrom/sol_9_x86/x86S0
rosemary (SPARC-system)# mount remote_x86_system_name:/cdrom/sol_9_x86/x86S2
rosemary (SPARC-system)# cd /x86S2
rosemary (SPARC-system)# ./modify_install_server -p /export/home/cdx86 /x86S0
```

In this example, each CD is inserted and automatically mounted before each of the commands. After each command, the CD is removed.

▼ To Create a SPARC Install Server on an x86 System With SPARC CD Media

Use this procedure to create a SPARC install server on an x86 system with SPARC CD media. You need the following:

- An x86 system
- A SPARC system with a CD-ROM drive
- A set of CDs for the remote SPARC system
  - Solaris Software 1 of 2 *SPARC Platform Edition* CD
  - Solaris Software 2 of 2 *SPARC Platform Edition* CD
  - Solaris *SPARC Platform Edition* Languages CD
  - Solaris *SPARC Platform Edition* Installation CD

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

In this procedure, *x86-system* is the x86 system that is to be the install server and *remote-SPARC-system* is the remote SPARC system to be used with the SPARC CD media.

1. **On the remote SPARC system, become superuser.**

   The system must include a CD-ROM drive and be part of the site's network and name service. If you use a name service, the system must also be in the NIS, NIS+, DNS, or LDAP name service. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **On the remote SPARC system, insert the Solaris Software 1 of 2** *SPARC Platform Edition* **CD into the system's drive.**

3. **On the remote SPARC system, add the following entries to the `/etc/dfs/dfstab` file.**

   ```
   share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
   share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
   ```

4. **On the remote SPARC system, start the NFS daemon.**

   ```
   remote-SPARC-system# /etc/init.d/nfs.server start
   ```

5. **On the remote SPARC system, verify that the CD is available to other systems by using the share command.**

   ```
   remote-SPARC-system# share
   -    /cdrom/cdrom0/s0 ro,anon=0 " "
   -    /cdrom/cdrom0/s1 ro,anon=0 " "
   ```

6. **On the x86 system that is to be the SPARC install server, become superuser.**

7. **On the x86 system, access the SPARC CD by creating two directories for the appropriate mount points, one for the miniroot and one for the product.**

   ```
   x86-system# mkdir directory_name_s0
   ```

   ```
   x86-system# mkdir  directory_name_s1
   ```

   directory_name_s0 Is the name of the directory to contain the product from slice 0

   directory_name_s1 Is the name of the directory to contain the miniroot from slice 1

8. **Verify that the CD is properly exported on the remote x86 system.**

```
x86-system# showmount -e remote-SPARC-system
export list for remote-SPARC-system:
/cdrom/sol_9_sparc/s0 (everyone)
/cdrom/sol_9_sparc/s1 (everyone)
```

9. **On the x86 system, mount the remote SPARC CD image.**

```
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s0 directory_name_s0
```

```
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s1 directory_name_s1
```

10. **On the x86 system, change to the `Tools` directory on the mounted disc:**

```
x86-system# cd /directory_name_s0/Solaris_9/Tools
```

11. **On the x86 system, copy the disc in the drive to the install server's hard disk in the directory you've created by using the `setup_install_server` command:**

```
x86-system# ./setup_install_server -t directory_name_s1 install_dir_path
```

`-t` Specifies the path to a boot image if you want to use a boot image other than the one in the `Tools` directory on the Solaris Software 2 of 2 CD.

directory_name_s1 Is the name of the directory that contains the miniroot from slice 1.

install_dir_path Specifies the directory where the disc image is to be copied. The directory must be empty.

Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

---

12. **On the x86 system, change to the top directory.**

```
x86-system# cd /
```

13. **On the x86 system, unmount both directories.**

```
x86-system# unmount directory_name_s0
```

```
x86-system# unmount directory_name_s1
```

14. **On the SPARC system, unshare both CD-ROM slices.**

```
remote-SPARC-system# unshare /cdrom/cdrom0/s0
```

```
remote-SPARC-system# unshare /cdrom/cdrom0/s2
```

15. **On the SPARC system, eject the Solaris Software 1 of 2** *SPARC Platform Edition* **CD.**

16. **Insert the Solaris Software 2 of 2** *SPARC Platform Edition* **CD into the x86 system's CD-ROM drive.**

17. **On the x86 system, change to the `Tools` directory on the mounted CD:**

    ```
    x86-system# cd /cdrom/cdrom0/Solaris_9/Tools
    ```

18. **On the x86 system, copy the CD to the install server's hard disk:**

    ```
    x86-system# ./add_to_install_server install_dir_path
    ```

    install_dir_path Specifies the directory where the CD image is to be copied

19. **Eject the Solaris Software 2 of 2** *SPARC Platform Edition* **CD.**

20. **On the x86 system, insert the Solaris** *SPARC Platform Edition* **Languages CD into the x86 system's CD-ROM drive and mount the CD.**

21. **On the x86 system, change to the `Tools` directory on the mounted CD:**

    ```
    x86-system# cd /cdrom/cdrom0/Tools
    ```

22. **On the x86 system, copy the CD to the install server's hard disk:**

    ```
    x86-system# ./add_to_install_server install_dir_path
    ```

    install_dir_path Specifies the directory where the CD image is to be copied

23. **Decide if you want to enable users to use the Solaris Web Start installation method to boot a system and install the Solaris 9 software from a network.**

    - If no, eject the Solaris *SPARC Platform Edition* Languages CD and proceed to Step 31.
    - If yes, eject the Solaris *SPARC Platform Edition* Languages CD and continue.

24. **On the remote SPARC system, insert the Solaris** *SPARC Platform Edition* **Installation CD into the SPARC system's CD-ROM drive.**

25. **On the remote SPARC system, add the following entry to the `/etc/dfs/dfstab` file.**

    ```
    share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
    share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
    ```

26. **On the remote SPARC system, start the NFS daemon.**

    ```
    remote-SPARC-system# /etc/init.d/nfs.server start
    ```

27. **On the remote SPARC system, verify that the CD is available to other systems:**

    ```
    remote-SPARC-system# share
    -   /cdrom/multi_sol_9_x86/s0 ro,anon=0 " "
    ```

```
-   /cdrom/multi_sol_9_x86/s1 ro,anon=0 " "
```

In the previous sample output, `multi_sol_9_x86` refers to the Solaris 9 (*x86 Platform Edition*) operating environment. This text string varies for each version of the Solaris operating environment.

28. **On the x86 system, mount the slices on the CD.**

```
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s0 directory_name_s0
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s1  directory_name_s1
```

29. **On the x86 system, change to the directory that contains `modify_install_server` on the mounted CD:**

```
x86-system# cd directory_name_s0
```

30. **Copy the Solaris Web Start interface software to the install server.**

```
x86-system# ./modify_install_server -p install_dir_path directory_name_s1
```

`-p` Preserves the existing image's miniroot in *install_dir_path*/`Solaris_9/Tools/Boot.orig`

install_dir_path Specifies the directory where the Solaris Web Start interface is to be copied

directory_name_s1 Name of the directory to contain the miniroot from slice 1

31. **Decide if you want to patch the files that are located in the miniroot (`Solaris_9/Tools/Boot`) on the net install image that was created by `setup_install_server`.**

- If no, proceed to the next step.
- If yes, use the `patchadd -C` command to patch the files that are located in the miniroot.

Caution – Don't use the `patchadd -C` unless you have read the `Patch README` instructions or have contacted your local Sun support office.

32. **Decide if you need to create a boot server.**

- If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. See "Adding Systems to Be Installed From the Network With a CD Image" on page 167.
- If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must create a boot server. For detailed instructions on how to create a boot server, refer to "To Create a Boot Server on a Subnet With a CD Image" on page 165.

EXAMPLE 15–4 Creating a SPARC Install Server on an x86 System With SPARC CD Media

The following example illustrates how to create a SPARC install server on an x86 system that is named `richards`. The following SPARC CDs are copied from a remote SPARC system that is named `simpson` to the x86 install server's `/export/home/cdsparc` directory.

- Solaris Software 1 of 2 *SPARC Platform Edition* CD
- Solaris Software 2 of 2 *SPARC Platform Edition* CD
- Solaris *SPARC Platform Edition* Languages CD
- Solaris *SPARC Platform Edition* Installation CD

On the remote SPARC system:

```
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
simpson (remote-SPARC-system)# /etc/init.d/nfs.server start
```

On the x86 system:

```
richards (x86-system)# mkdir /sparcS0
richards (x86-system)# mkdir /sparcS1
richards (x86-system)# mount simpson:/cdrom/cdrom0/s0 /sparcS0
richards (x86-system)# mount simpson:/cdrom/cdrom0/s1 /sparcS1
richards (x86-system)# cd /sparcS0/Solaris_9/Tools
richards (x86-system)# ./setup_install_server -t /sparcS0 /export/home/cdsparc
richards (x86-system)# cd /
richards (x86-system)# unmount /sparcS0
richards (x86-system)# unmount /sparcS1
```

On the remote SPARC system:

```
simpson (remote-SPARC-system) unshare /cdrom/cdrom0/s0
simpson (remote-SPARC-system) unshare  /cdrom/cdrom0/s1
```

On the x86 system:

```
richards (x86-system)# cd /cdrom/cdrom0/Solaris_9/Tools
richards (x86-system)# ./add_to_install_server /export/home/cdsparc
richards (x86-system)# cd /cdrom/cdrom0/Tools
richards (x86-system)# ./add_to_install_server /export/home/cdsparc
```

On the remote SPARC system:

```
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
simpson (remote-SPARC-system)# /etc/init.d/nfs.server start
```

On the x86 system:

```
richards (x86-system)# mount remote_SPARC_system_name:/cdrom/cdrom0 /sparcS0
richards (x86-system)# mount remote_SPARC_system_name:/cdrom/cdrom0 /sparcS1
richards (x86-system)# cd /sparcS0
richards (x86-system)# ./modify_install_server -p /export/home/cdsparc /sparcS1
```

EXAMPLE 15–4 Creating a SPARC Install Server on an x86 System With SPARC CD Media *(Continued)*

In this example, each CD is inserted and automatically mounted before each of the commands. After each command, the CD is removed.

Creating a Boot Server on a Subnet With a CD Image

You must create an install server to install the Solaris software on a system from the network. You do not always need to set up a boot server. A boot server contains enough of the boot software to boot systems from the network, and then the install server completes the installation of the Solaris software.

- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a boot server. Proceed to "Adding Systems to Be Installed From the Network With a CD Image" on page 167.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.

▼ To Create a Boot Server on a Subnet With a CD Image

1. **On the system you intend to make the boot server for the subnet, log in and become superuser.**

   The system must include a local CD-ROM drive or have access to the remote Solaris 9 disc images, which are normally on the install server. If you use a name service, the system should be in the name service. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Mount the Solaris Software 1 of 2 CD image from the install server.**

   ```
   # mount -F nfs -o ro server_name:path /mnt
   ```

   server_name:*path* Is the install server name and absolute path to the disc image

3. **Create a directory for the boot image.**

   ```
   # mkdir -p boot_dir_path
   ```

boot_dir_path	Specifies the directory where the boot software is to be copied

4. **Change to the `Tools` directory on the Solaris Software 1 of 2 CD image.**

   ```
   # cd /mnt/Solaris_9/Tools
   ```

5. **Copy the boot software to the boot server.**

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

`-b`	Specifies to set up the system as a boot server
boot_dir_path	Specifies the directory where the boot software is to be copied

   ---

   Note – The `setup_install_server` command indicates whether you have enough disk space available for the images. To determine available disk space, use the `df -kl` command.

   ---

6. **Change directories to root (/).**

   ```
   # cd /
   ```

7. **Unmount the installation image.**

   ```
   # umount /mnt
   ```

You are now ready to set up systems to be installed from the network. See “Adding Systems to Be Installed From the Network With a CD Image” on page 167.

EXAMPLE 15–5 Creating a Boot Server on a Subnet With CD Media

The following example illustrates how to create a boot server on a subnet. These commands copy the boot software from the Solaris Software 1 of 2 *SPARC Platform Edition* CD image to `/export/install/boot` on the system’s local disk.

```
# mount -F nfs -o ro crystal:/export/install/boot /mnt
# mkdir -p /export/install/boot
# cd /mnt/Solaris_9/Tools
# ./setup_install_server -b /export/install/boot
# cd /
# umount /mnt
```

In this example, the disc is inserted and automatically mounted before the command. After the command, the disc is removed.

Adding Systems to Be Installed From the Network With a CD Image

After you create an install server and, if necessary, a boot server, you must set up each system that you want to install from the network. Each system that you want to install needs to find the following:

- An install server
- A boot server if it is required
- The `sysidcfg` file if you use a `sysidcfg` file to preconfigure system information
- A name server if you use a name service to preconfigure system information
- The profile in the JumpStart directory on the profile server if you are using the custom JumpStart installation method

Use the following `add_install_client` procedure for setting up install servers and clients. Also, see the example procedures for the following:

- If you are using DHCP to set installation parameters, see Example 15–6.
- If your install server and client are on the same subnet, see Example 15–7.
- If your install server and your client are not on the same subnet and you are not using DHCP, see Example 15–8.
- If you are using DHCP to set installation parameters for x86 clients, see Example 15–9.
- If you want to use a specific serial port to display output during the installation of an x86-based system, see Example 15–10.
- If you want to set up an x86 client to use a specific network interface during the installation, see Example 15–11.

For more options to use with this command, see the man page, `add_install_client`(1M).

▼ To Add Systems to Be Installed From the Network With `add_install_client` (CDs)

If you have a boot server, make sure you have shared the install server installation image. See the procedure "To Create an Install Server," Step 6.

1. **On the install server or boot server, become superuser.**

2. **If you use the NIS, NIS+, DNS, or LDAP name service, verify that the following information about the system to be installed has been added to the name service:**

   - Host name
   - IP address
   - Ethernet address

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

3. **Change to the `Tools` directory on the Solaris 9 CD image on the install server:**

   ```
   # cd /install_dir_path/Solaris_9/Tools
   ```

   install_dir_path Specifies the path to the `Tools` directory

4. **Set up the client system to be installed from the network.**

   ```
   # ./add_install_client [-d] [-s install_server:install_dir_path] \
   [-c jumpstart_server:jumpstart_dir_path]     [-p sysid_server:path] \
     [-t boot_image_path] [-b "network_boot_variable=value"] \
   [-e ethernet_address] client_name platform_group
   ```

   `-d` Specifies that the client is to use DHCP to obtain the network install parameters. If you use the `-d` only, the `add_install_client` command sets up the installation information for client systems of the same class, for example, all SPARC client machines. To set up the installation information for a specific client, use the `-d` with the `-e` option.

   For x86 clients, use this option to boot the systems from the network by using PXE network boot.

   For more information about class-specific installations by using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86.

   `-s` *install_server:install_dir_path* Specifies the name and path to the install server.

   - *install_server* is the host name of the install server
   - *install_dir_path* is the absolute path to the Solaris 9 CD image

   `-c` *jumpstart_server* : *jumpstart_dir_path* Specifies a JumpStart directory for custom JumpStart installations.

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

`-p` *sysid_server*`:`*path*

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

`-t` *boot_image_path*

Specifies the path to an alternate boot image if you want to use a boot image other than the one in the Tools directory on the Solaris 9 net installation image, CD, or DVD.

`-b` *"boot-property=value"*

x86 based systems only: Enables you to set the value of a boot property variable that you want to use to boot the client from the network. The `-b` must be used with the `-e` option.

See the `eeprom`(1M) man page for descriptions of boot properties.

`-e` *ethernet_address*

Specifies the ethernet address of the client that you want to install. This option enables you to set up the installation information to use for a specific client.

For more information about client-specific installations by using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86.

client_name

Is the name of the system to be installed from the network. This name is *not* the host name of the install server.

platform_group

Is the platform group of the system to be installed. A detailed list of platform groups appears in Appendix C.

EXAMPLE 15–6 SPARC: Adding a SPARC Install Client on a SPARC Install Server When Using DHCP (CDs)

The following example illustrates how to add an install client when you are using DHCP to set installation parameters on the network. The install client is named `basil`, which is an Ultra 5 system. The file system `/export/home/cdsparc/Solaris_9/Tools` contains the `add_install_client` command.

For more information on how to use DHCP to set installation parameters for network installations, see "Supporting Solaris Network Installation with the DHCP Service (Task Map)" in *System Administration Guide: IP Services*.

```
sparc_install_server# cd /export/home/cdsparc/Solaris_9/Tools
sparc_install_server# ./add_install_client -d basil sun4u
```

EXAMPLE 15–7 Adding an Install Client That Is on the Same Subnet as Its Server (CDs)

The following example illustrates how to add an install client that is on the same subnet as the install server. The install client is named `basil`, which is an Ultra 5 system. The file system `/export/home/cdsparc/Solaris_9/Tools` contains the `add_install_client` command.

```
install_server# cd /export/home/cdsparc/Solaris_9/Tools
install_server# ./add_install_client basil sun4u
```

EXAMPLE 15–8 Adding an Install Client to a Boot Server (CDs)

The following example illustrates how to add an install client to a boot server. The install client is named `rose`, which is an Ultra 5 system. Run the command on the boot server. The `-s` option is used to specify an install server that is named `rosemary`, which contains a Solaris 9 CD image in `/export/home/cdsparc`.

```
boot_server# cd /export/home/cdsparc/Solaris_9/Tools
boot_server# ./add_install_client -s rosemary:/export/home/cdsparc rose sun4u
```

EXAMPLE 15–9 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP (CDs)

The following example illustrates how to add an x86 install client to an install server when you are using DHCP to set installation parameters on the network. The `-d` option is used to specify that clients are to use the DHCP protocol for configuration. If you plan to use PXE network boot, you must use the DHCP protocol. The DHCP class name SUNW.i86pc indicates that this command applies to all Solaris x86 network boot clients, not just a single client. The `-s` option is used to specify that the clients are to be installed from the install server that is named `rosemary`. This server contains a Solaris Software 1 of 2 *x86 Platform Edition* CD image in `/export/home/cdx86`.

EXAMPLE 15–9 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP (CDs) *(Continued)*

For more information on how to use DHCP to set installation parameters for network installations, see "Supporting Solaris Network Installation with the DHCP Service (Task Map)" in *System Administration Guide: IP Services*.

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -s rosemary:/export/home/cdx86 SUNW.\
i86pc i86pc
```

EXAMPLE 15–10 x86: Specifying a Serial Console to Use During a Network Installation (CDs)

The following example illustrates how to add an x86 install client to an install server and specify a serial console to use during the installation. This example sets up the install client in the following manner.

- The `-d` option indicates that the client is set up to use DHCP to set installation parameters.
- The `-e` option indicates that this installation will occur only on the client with the ethernet address 00:07:e9:04:4a:bf.
- The first and second uses of the `-b` option instruct the installation program to use the serial port `ttya` as an input and an output device.

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" i86pc
```

For a complete description of the boot property variables and values you can use with the `-b` option, see the `eeprom`(1M) man page.

EXAMPLE 15–11 x86: Specifying a Boot Device to Use During a Network Installation (CDs)

The following example illustrates how to add an x86 install client to an install server and specify a boot device to use during the installation. If you specify the boot device when you set up the install client, you are not prompted for this information by the Device Configuration Assistant during the installation.

This example sets up the install client in the following manner.

- The `-d` option indicates that the client is set up to use DHCP to set installation parameters.
- The `-e` option indicates that this installation will occur only on the client with the ethernet address 00:07:e9:04:4a:bf.
- The first and second uses of the `-b` option instruct the installation program to use the serial port `ttya` as an input and an output device.
- The third use of the `-b` option instructs the installation program to use a specific boot device during the installation.

Note – The value of the boot device path varies based on your hardware.

- The i86pc platform name indicates that the client is an x86-based system.

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

For a complete description of the boot property variables and values you can use with the `-b` option, see the `eeprom`(1M) man page.

CHAPTER 16

Preparing to Install From the Network (Command Reference)

This chapter lists the commands used to set up network installations.

TABLE 16–1 Network Installation Commands

Command	Platform	Description
`add_install_client`	All	A command that adds network installation information about a system to an install server or boot server from the network. The `add_install_client`(1M) man page contains more information.
`setup_install_server`	All	A script that copies the Solaris 9 DVD or CDs to an install server's local disk or copies the boot software to a boot server. The `setup_install_server`(1M) man page contains more information.
(CD media only) `add_to_install_server`	All	A script that copies additional packages within a product tree on the CDs to the local disk on an existing install server. The `add_to_install_server`(1M) man page contains more information.
(CD media only) `modify_install_server`	All	A script that adds the Solaris Web Start user interface software to the Solaris 9 CD images on an existing install server or boot server. This script enables you to use the Solaris Web Start program to boot a system and install the Solaris 9 software from a network. The `modify_install_server`(1M) man page contains more information. A system with 64 Mbytes or less RAM does not have enough memory to install a machine by using the Solaris Web Start program.
`mount`	All	A command that enables the mounting of file systems and shows the mounted file systems, including the file system on the Solaris DVD or Solaris Software and Solaris Languages CDs. The `mount`(1M) man page contains more information.

TABLE 16–1 Network Installation Commands *(Continued)*

Command	Platform	Description
`showmount -e`	All	A command that lists all the shared file systems that are located on a remote host. The `showmount`(1M) man page contains more information.
`uname -i`	All	A command for determining a system's platform name, for example, SUNW, SPARCstation-5, or i86pc. You might need the system's platform name when you install the Solaris software. The `uname`(1) man page contains more information.
`patchadd -C` *net_install_image*	All	A command to add patches to the files that are located in the miniroot, `Solaris_9/Tools/Boot`, on a net installation image of a DVD or CD that is created by `setup_install_server`. This facility enables you to patch Solaris installation commands and other miniroot-specific commands. *net_install_image* is the absolute path name of the net installation image. The `patchadd`(1M) man page contains more information. **Caution –** Don't use the `patchadd -C` command unless you have read the `Patch README` instructions or have contacted your local Sun support office.
`reset`	SPARC	An Open Boot PROM command for resetting the system and rebooting the machine. Or, if you boot and see a series of error messages about I/O interrupts, press the Stop and A keys at the same time, and then type `reset` at the `ok` or `>` PROM prompt.
`banner`	SPARC	An Open Boot PROM command that displays system information, such as model name, Ethernet address, and memory installed. You can issue this command only at the `ok` or `>` PROM prompt.

CHAPTER **17**

Using the Solaris Web Start Program (Tasks)

This chapter explains how to use the Solaris Web Start program on the Solaris DVD or Solaris Installation CD to install or upgrade Solaris software.

Note – If you want to install the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive, you can use a DVD-ROM or CD-ROM drive that is attached to another machine. For detailed instructions, refer to Appendix F.

This chapter contains the following topics:

Solaris Web Start Program GUI or CLI

You can run the Solaris Web Start program with a GUI or with a CLI. For x86 systems, the Device Configuration Assistant is included in the Solaris Web Start program.

- GUI – Requires a local or remote DVD-ROM or CD-ROM drive or network connection, video adapter, keyboard, monitor, and enough memory.
- CLI – Requires a local or remote DVD-ROM or CD-ROM drive or network connection, keyboard, and monitor. You can run the Solaris Web Start installation CLI with the `tip` command. For more information, see the man page, `tip`(1).

If the Solaris Web Start program detects a video adapter for the system, it automatically displays the GUI. If the Solaris Web Start program does not detect a video adapter, it automatically displays the CLI. The content and sequence of instructions in both the GUI and the CLI are generally the same.

SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program

You can either install or upgrade the Solaris operating environment by using the Solaris Web Start program, which is on the Solaris *SPARC Platform Edition* DVD or the Solaris *SPARC Platform Edition* Installation CD.

Note – If you are installing from the Solaris Installation CD, refer to "Requirements When Using the Solaris Installation CD" on page 46 for special requirements.

For the installation, ensure that you have the following:

- For a DVD installation, the Solaris *SPARC Platform Edition* DVD
- For a CD installation:
  - Solaris *SPARC Platform Edition* Installation CD.
  - Solaris Software 1 of 2 *SPARC Platform Edition* CD.
  - Solaris Software 2 of 2 *SPARC Platform Edition* CD – The installation program prompts you for this CD if necessary.
  - Solaris *SPARC Platform Edition* Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

SPARC: Task Map: Performing a Solaris Web Start Installation

TABLE 17–1 Performing a Solaris Web Start Installation Task Map

Task	Description	For Instructions
Verify system requirements.	Verify that your system meets the requirements to install or upgrade with the Solaris Web Start program.	"Memory Requirements" on page 43
Gather the necessary information.	Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.	Chapter 6
(Optional) Preconfigure system configuration information.	Use the `sysidcfg` file or the name service to preconfigure installation information (for example, `locale`) for a system. Preconfiguring system information prevents the installation program from prompting you during the installation.	Chapter 7
(Upgrade only) Prepare to upgrade the system.	Back up the system.	*System Administration Guide: Devices and File Systems*
(Optional) Set up the system to install from the network.	To install a system from a remote DVD or CD net installation image, you need to set up the system to boot and install from an install server or boot server.	Chapter 15
Install or upgrade.	Boot the system and follow the prompts to install or upgrade the Solaris software.	"SPARC: To Perform an Installation or Upgrade With the Solaris Web Start Program" on page 177
(Upgrade only) Perform post-upgrade tasks.	Correct any local modifications that were lost during the upgrade.	"To Correct Local Modifications After Upgrading" on page 189

▼ SPARC: To Perform an Installation or Upgrade With the Solaris Web Start Program

1. **Decide if you want to install the software by using the DVD-ROM or CD-ROM drive or by using a net image.**
   - If you're using a DVD-ROM or CD-ROM drive, insert the Solaris *SPARC Platform Edition* DVD or Solaris *SPARC Platform Edition* Installation CD.
   - If you're using a net installation image, change directories to where the installation media is located. You might need to contact your network administrator for the location. For more information on installing from the network, see Chapter 14 or Chapter 15.

2. **Boot the system.**

- If the system is new, out-of-the-box, turn on the system.
- If you want to install or upgrade an existing system, shut down the system.
  - To boot from the local DVD or CD, type the following command.

    ```
    ok boot cdrom [- nowin]
    ```

  - To boot from an install server on a network, type the following command.

    ```
    ok boot net [- nowin]
    ```

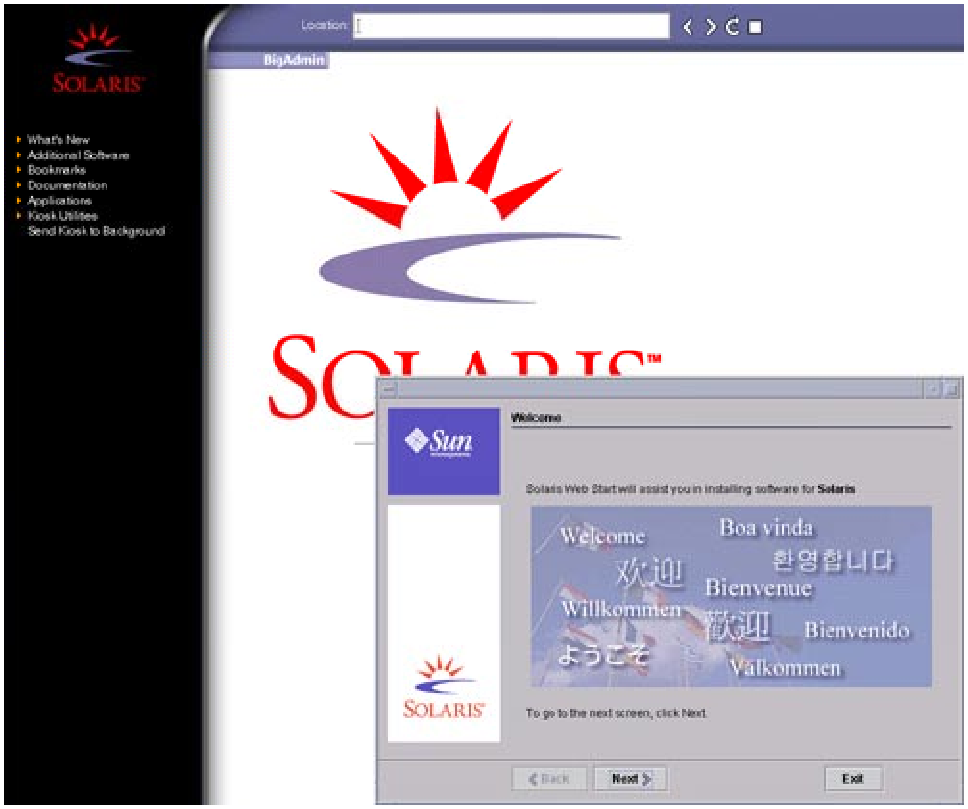
`nowin` Specifies to run the Solaris Web Start program in CLI mode.

The Solaris Web Start installation begins.

3. **If you are prompted, answer the system configuration questions.**

   - If you preconfigured all of the system configuration information, the Solaris Web Start program does not prompt you to enter any configuration information.
   - If you did not preconfigure the system configuration information, use the "Worksheet for Installation" on page 52 or the "Worksheet for Upgrading" on page 58 to help you answer the system configuration questions.

   If you are using the GUI, after you confirm the system configuration information, the Solaris Web Start Installation Kiosk and Welcome to Solaris dialog box appear. If your system has insufficient memory, the Kiosk does not display.



Solaris Web Start will assist you in installing software for **Solaris**

To go to the next screen, click Next.

FIGURE 17–1 Solaris Web Start Kiosk Menu

You can click on any link in the Kiosk menu.

Note – In some situations, the Kiosk might cover a dialog box. To display a hidden dialog box, from the Kiosk menu, choose Send Kiosk to Background.

The Installer Questions screen appears.

4. **Decide if you want to reboot the system automatically and if you want to automatically eject the disc. Click NEXT.**

   The Specify Media screen appears.

5. **Specify the media you are using to install.**

   - If you are installing a Solaris Flash archive, continue.
   - If you are *not* installing a Solaris Flash archive, proceed to Step 6.

a. **Type the information that you are prompted to enter.**

Media Selected	Prompt
DVD or CD	Insert the disc where the Solaris Flash archive is located.
Network File System	Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.
HTTP	Specify the URL and proxy information that is needed to access the Solaris Flash archive.
FTP	Specify the FTP server and the path to the Solaris Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.
Local tape	Specify the local tape device and the position on the tape where the Solaris Flash archive is located.

If you selected to install an archive from a DVD, CD, or from an NFS server, the Select Flash Archives panel is displayed.

b. **For archives that are stored on a disc or an NFS server, on the Select Flash Archives panel, select one or more Solaris Flash archives to install.**

c. **On the Flash Archives Summary panel, confirm the selected archives and click Next.**

d. **On the Additional Flash Archives panel, you can install an additional Solaris Flash archive by specifying the media where the other archive is located. If you do not want to install additional archives, select None and click Next to continue the installation. Proceed to Step 7.**

6. **Decide if you want to perform an initial installation or an upgrade.**

   The Solaris Web Start program determines if the system can be upgraded. You must have an existing Solaris root (/) file system. If you are upgrading by using the Solaris Installation CD, you must have a 512–Mbyte slice. The Solaris Web Start program detects the necessary conditions and then upgrades the system.

7. **Follow the instructions on the screen to install the Solaris software. If you are NOT installing a Solaris Flash archive, you can also install additional software.**

   When the Solaris Web Start program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

   After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

   - `/var/sadm/system/logs`
   - `/var/sadm/install/logs`

8. **If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to "To Correct Local Modifications After Upgrading" on page 189.**

x86: Performing an Installation or Upgrade With the Solaris Web Start Program

You can install the Solaris operating environment on your x86 system by using the Solaris Web Start program on the Solaris DVD or Solaris Installation CD.

Note – If you are installing from the Solaris Installation CD, refer to "Requirements When Using the Solaris Installation CD" on page 46 for special requirements.

Ensure that you have the following media:

- If you are installing from a DVD, use the Solaris *x86 Platform Edition* DVD
- If you are installing from CD media, use the following:
  - Solaris *x86 Platform Edition* Installation CD
  - Solaris Software 1 of 2 *x86 Platform Edition* CD
  - Solaris Software 2 of 2 *x86 Platform Edition* CD – The installation program prompts you for this CD if necessary.
  - Solaris *x86 Platform Edition* Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

x86: Task Map: Performing a Solaris Web Start Installation

TABLE 17–2 x86: Performing a Solaris Web Start Installation Task Map

Task	Description	For Instructions
Verify system requirements.	Verify that your system meets the requirements to install or upgrade with the Solaris Web Start program.	"Memory Requirements" on page 43

TABLE 17–2 x86: Performing a Solaris Web Start Installation Task Map *(Continued)*

Task	Description	For Instructions
Gather the necessary information.	Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.	Chapter 6
(Upgrade only) Prepare to upgrade the system.	Back up the system.	*System Administration Guide: Devices and File Systems*
(Optional) Set up the system to install from the network.	To install a system from a remote Solaris *x86 Platform Edition* DVD or Solaris Software *x86 Platform Edition* CD net installation image, you must set up the system to boot and install from an install server or boot server.	Chapter 15
Install or upgrade.	Boot the system and follow the prompts to install or upgrade the Solaris software.	"x86: To Perform an Installation or Upgrade With the Solaris Web Start Program" on page 182
(Upgrade only) Perform post-upgrade task.	Correct any local modifications that were lost during the upgrade.	"To Correct Local Modifications After Upgrading" on page 189

▼ x86: To Perform an Installation or Upgrade With the Solaris Web Start Program

1. **Decide if you want to install the software by using the DVD-ROM or CD-ROM drive or by using a net installation image.**

   - If you're using a DVD-ROM or CD-ROM drive, insert the Solaris *x86 Platform Edition* DVD or the Solaris *x86 Platform Edition* Installation CD.
   - If you're using a net installation image, change directories to where the installation media is located. You might need to check with your network administrator for the location. For more information on installing from the network, see Chapter 14 or Chapter 15.

2. **Decide how to boot the system.**

   - If you boot from the Solaris DVD or the Solaris Installation CD, insert the disc. Your system's BIOS must support booting from a DVD or CD.
   - If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the system's BIOS setup tool or the network adapter's configuration setup tool.
   - If you boot from a diskette, insert a Solaris Device Configuration Assistant *x86 Platform Edition* diskette into the system's diskette drive.

Note – You can copy the Device Configuration Assistant software to a diskette from the Solaris *x86 Platform Edition* DVD or Solaris Software 2 of 2 *x86 Platform Edition* CD by using the procedure in Appendix I.

3. **Boot the system by shutting it down and then turning it off and on.**

   A memory test and hardware detection are executed. The screen refreshes.

4. **When the Solaris Device Configuration Assistant screen is displayed, press F2_Continue.**

   The Bus Enumeration screen appears with the message:

   ```
   Determining bus types and gathering hardware configuration data ...
   ```

   The Scanning Devices screen appears. System devices are scanned. When scanning is complete, the Identified Devices screen appears.

5. **Decide if you need to make any changes.**

   - Press F2_Continue to make no changes.
   - Select changes and press F4.

   The Loading screen is displayed with messages about drivers that are loaded to boot your system. After a few seconds, the Boot Solaris screen appears.

6. **On the Boot Solaris screen, select DVD, CD, Net, or Disk, and press F2_Continue.**

   The Solaris Web Start program checks the default boot disk for the requirements to install or upgrade the system.

Note – If you are installing by using the Solaris Installation CD, several requirements must be met. If you cannot meet these requirements, use the Solaris Web Start program from the DVD or a net installation image, or the Solaris `suninstall` program or custom JumpStart to install. These requirements are the following:

- The BIOS and SCSI driver for the default boot disk must support logical block addressing (LBA).
- If the default boot disk does not have a Solaris `fdisk` partition, you are prompted to create a partition. You are asked if you want to run `fdisk` and create a Solaris `fdisk` partition. If you answer Yes, the `fdisk` user interface opens so you can manually add a Solaris `fdisk` partition on the disk. If you answer No, the installer exits.
- If the default boot disk does not have an x86 boot partition, the Solaris Web Start program creates a partition for you from a portion of the Solaris `fdisk` partition. Any data on the Solaris `fdisk` partition is destroyed. You cannot create this partition manually. If you answer Yes to formatting the disk, the partition is created from the Solaris `fdisk` partition. If you answer No, you must choose another method of installing.
- If you want to preserve an existing Service partition on your system, select Default when the Solaris Web Start program asks you if you want to customize the `fdisk` partitions on the default boot disk.
- Later in the installation, you can create, modify, or delete partitions through format panels. But, after the Solaris `fdisk` partition and x86 boot partition are created, the disk is frozen and you cannot edit the disk. If you have multiple disks, you can edit those disks at the `fdisk` panels.

The Solaris Web Start program detects each requirement on the default boot disk and prompts you for configuration information that was not found.

7. **If you are prompted, answer the system configuration questions.**
   - If you preconfigured all of the system configuration information, proceed to Step 8.
   - If you did not preconfigure the system configuration information, use the "Worksheet for Installation" on page 52 or the "Worksheet for Upgrading" on page 58 to help you answer the system configuration questions.

   After a few seconds, the Solaris Installation Program screen appears.

8. **On the Solaris Installation Program screen, press F2_Continue.**

   The kdmconfig – Introduction screen appears.

9. **Decide if you have enough memory to run the GUI.**
   - Press F2_Continue if you have enough memory to display the GUI.

- Press F4_Bypass if you don't have enough memory, and the CLI is displayed.

The kdmconfig – View and Edit Window System Configuration screen appears.

If the `kdmconfig` utility cannot detect the video driver, the `kdmconfig` utility selects the 640x480 VGA driver. The Solaris Web Start GUI cannot be displayed with the 640x480 VGA driver. As a result, the Solaris Web Start CLI is displayed. To use the Solaris Web Start GUI, use the `kdmconfig` utility to select the correct video driver for your system.

10. **(Optional) Examine the configuration information on the kdmconfig – View and Edit Window System Configuration screen and make any changes you need.**

11. **When you are finished, select No changes needed – Test/Save and Exit and press F2_Continue.**

    The kdmconfig Window System Configuration Test screen appears.

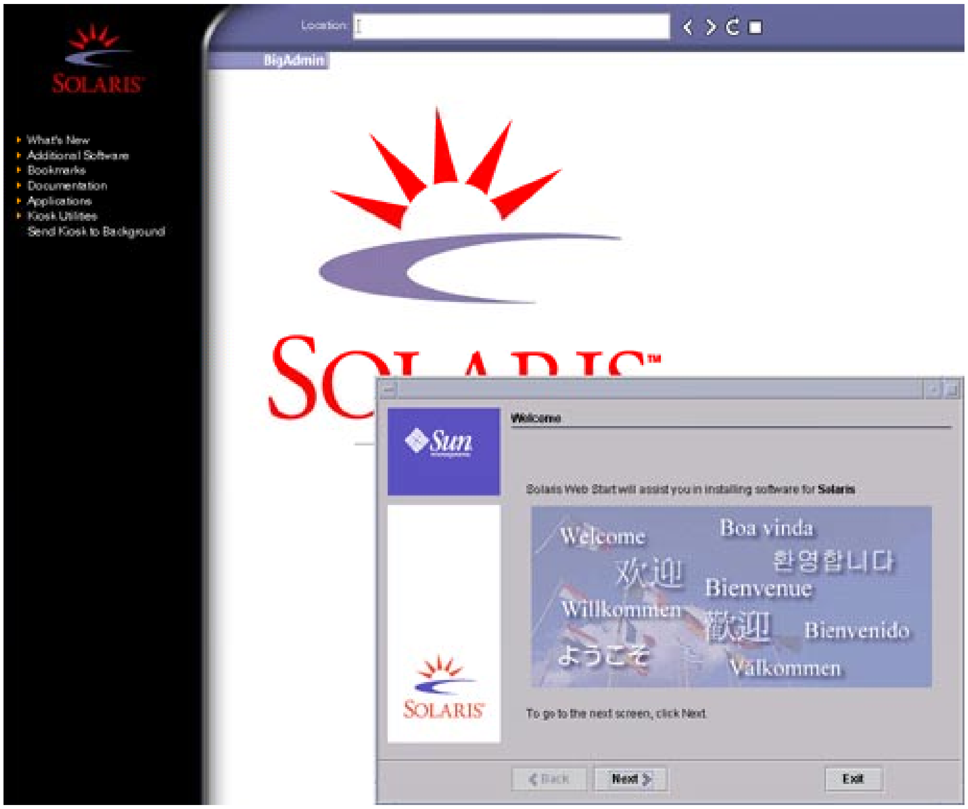
12. **Press F2_Continue.**

    The screen refreshes and the kdmconfig Window System Configuration Test palette and pattern screen appears.

13. **Move the pointer and examine the colors that are shown on the palette to ensure that they are displayed accurately.**

    - If the colors are not displayed accurately, click No. If possible, press any key on the keyboard, or wait until `kdmconfig` exits the kdmconfig Window System Configuration Test screen automatically. Repeat Step 10 through Step 13 until the colors are displayed accurately and you can move the pointer as expected.
    - If the colors are displayed accurately, click Yes.

    The Solaris Web Start Installation Kiosk and Welcome to Solaris dialog box appear. If your system has insufficient memory, the Kiosk is not displayed.



Solaris Web Start will assist you in installing software for **Solaris**

To go to the next screen, click Next.

FIGURE 17–2 Solaris Web Start Kiosk Menu

You can click on any link in the Kiosk menu.

Note – In some situations, the Kiosk might cover a dialog box. To display a hidden dialog box, from the Kiosk menu, choose Send Kiosk to Background.

---

14. If you are prompted, answer any remaining configuration questions.

The Installer Questions screen appears.

15. Decide if you want to reboot the system automatically and if you want to automatically eject the disc. Click NEXT.

Note – If you are using PXE to boot from a network, select manual reboot. You must ensure that the system does not boot from the network when it reboots. To disable network boot, during the reboot use the system's BIOS setup tool or network adapter's configuration setup tool.

The Specify Media screen appears.

16. **Specify the media you are using to install.**

- If you are installing a Solaris Flash archive, continue.
- If you are *not* installing a Solaris Flash archive, proceed to Step 17.

a. **Type the information that you are prompted to enter.**

Media Selected	Prompt
DVD or CD	Insert the disc where the Solaris Flash archive is located.
Network File System	Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.
HTTP	Specify the URL and proxy information that is needed to access the Solaris Flash archive.
FTP	Specify the FTP server and the path to the Solaris Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.
Local tape	Specify the local tape device and the position on the tape where the Solaris Flash archive is located.

If you selected to install an archive from a DVD, CD, or from an NFS server, the Select Flash Archives panel is displayed.

b. **For archives that are stored on a disc or an NFS server, on the Select Flash Archives panel, select one or more Solaris Flash archives to install.**

c. **On the Flash Archives Summary panel, confirm the selected archives and click Next.**

d. **On the Additional Flash Archives panel, you can install an additional Solaris Flash archive by specifying the media where the other archive is located. If you do not want to install additional archives, select None and click Next to continue the installation. Complete the installation.**

When the Solaris Web Start installation program is finished installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

- `/var/sadm/system/logs`
- `/var/sadm/install/logs`

17. **Decide if you want to perform an initial installation or an upgrade.**

    The Solaris Web Start program determines if the system can be upgraded. You must have an existing Solaris root (/) file system. If you are upgrading by using the Solaris Installation CD, you must have a 512–Mbyte slice. The Solaris Web Start program detects the necessary conditions and then upgrades the system.

    Note – If you restored the Service partition on your system before you started the installation, you might not be able to upgrade to the Solaris operating environment.

    ---

18. **Follow the instructions on the screen to install the Solaris software and any additional software on the system.**

    When the Solaris Web Start installation program is finished installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

    After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

    - `/var/sadm/system/logs`
    - `/var/sadm/install/logs`

19. **If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to “To Correct Local Modifications After Upgrading” on page 189.**

Solaris Web Start Postinstallation and Upgrade Tasks

After you install or upgrade the Solaris operating environment, you might need to perform these tasks.

▼ To Correct Local Modifications After Upgrading

When you upgrade, the Solaris Web Start program merges local software modifications of the existing system with the new Solaris software. However, in some situations, merging is not possible. After you finish upgrading a system, you might need to correct some local modifications that were not preserved.

Caution – Your system might not boot if you do not correct the local modifications that were not preserved.

1. **Review the contents of the following file to determine whether you need to correct local modifications that the Solaris Web Start program could not preserve.**

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

2. **Correct any local modifications that were not preserved.**

3. **Reboot the system.**

   ```
   # reboot
   ```

CHAPTER 18

Using the Solaris suninstall Program (Tasks)

This chapter explains how to use the Solaris suninstall program on the Solaris Software 1 of 2 CD to install or upgrade the Solaris software.

Solaris suninstall Program

You can run the Solaris suninstall program with a CLI. For x86 systems, the Device Configuration Assistant is included in the Solaris suninstall program. The Solaris suninstall program requires a local or remote CD-ROM drive or network connection, keyboard, and monitor. You can run the Solaris suninstall program with the tip command. For more information, see the man page, tip(1).

Note – To navigate through the text, use Control-U to move up a page and Control-D to move down a page.

SPARC: Performing an Installation or Upgrade With the Solaris `suninstall` Program

You can either install or upgrade the Solaris operating environment on your SPARC system by using the Solaris `suninstall` program on the Solaris Software 1 of 2 CD. You cannot install any additional software with the Solaris `suninstall` program.

Ensure that you have the following CDs:

- Solaris Software 1 of 2 *SPARC Platform Edition* CD.
- Solaris Software 2 of 2 *SPARC Platform Edition* CD – The installation program prompts you for this CD if necessary.
- Solaris *SPARC Platform Edition* Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

SPARC: Task Map: Performing a Solaris `suninstall` Program Installation

TABLE 18–1 SPARC: Task Map: Performing a Solaris `suninstall` Program Installation

Task	Description	For Instructions
Gather the necessary information.	Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.	Chapter 6
(Upgrade only) Prepare to upgrade the system.	Back up the system.	*System Administration Guide: Devices and File Systems*
(Optional) Preconfigure system configuration information.	Use the `sysidcfg` file or the name service to preconfigure installation information for a system, for example `locale`. Preconfiguring system information prevents the installation program from prompting you to supply the information during the installation.	Chapter 7
(Optional) Set up the system to install from the network.	To install a system from a remote Solaris Software *SPARC Platform Edition* CD image, you need to set up the system to boot and install from an install server or boot server.	Chapter 15

TABLE 18–1 SPARC: Task Map: Performing a Solaris suninstall Program Installation *(Continued)*

Task	Description	For Instructions
Install or upgrade.	Boot the system and follow the prompts to install or upgrade the Solaris software.	"SPARC: To Perform an Installation or Upgrade With the Solaris suninstall Program" on page 193
(Upgrade only) Perform post-upgrade task.	Correct any local modifications that were lost during the upgrade.	"Solaris suninstall Program Post-Upgrade Task" on page 201

▼ SPARC: To Perform an Installation or Upgrade With the Solaris suninstall Program

1. **Decide if you want to install the software by using a CD-ROM drive or by using a net image.**

   - If you're using a CD-ROM drive, insert the Solaris Software 1 of 2 *SPARC Platform Edition* CD
   - If you're using a net installation image, change directories to where the installation media is located, as in the following example. You might need to check with your network administrator for the location. The following command is an example.

     ```
     % cd /net/install-svr/export/Solaris_9/sparc
     ```

2. **Boot the system.**

   - If the system is new, out-of-the-box, turn on the system.
   - If you want to install or upgrade an existing system, shut down the system.
     - To boot from the local CD, type the following command.

       ```
       ok boot cdrom
       ```

     - To boot from an install server on a network, type the following command.

       ```
       ok boot net
       ```

   The Solaris suninstall program begins.

3. **If you are prompted, answer the system configuration questions.**

   If you preconfigured all of the system configuration information, the Solaris suninstall program does not prompt you to enter any configuration information.

   If you did not preconfigure the system configuration information, use the "Worksheet for Installation" on page 52 or the "Worksheet for Upgrading" on page 58 to help you answer the system configuration questions.

4. **Follow the instructions on the screen to install the Solaris software on the system.**

   - If you are *not* installing a Solaris Flash archive, complete the installation. If you are upgrading or need to install additional software, proceed to Step 5.

     When the Solaris `suninstall` program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

     After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories.

     - `/var/sadm/system/logs`
     - `/var/sadm/install/logs`
   - If you are installing a Solaris Flash archive, continue.

   a. **On the Flash Archive Retrieval Method screen, select the location of the Solaris Flash archive.**

   The Solaris `suninstall` program installation prompts you to proceed, depending on the media you selected.

   b. **Type the information that you are prompted to enter.**

Media Selected	Prompt
HTTP	Specify the URL and proxy information that is needed to access the Solaris Flash archive.
FTP	Specify the FTP server and the path to the Solaris Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.
Network File System	Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.
Local file	Specify the path to the local file system where the Solaris Flash archive is located.
Local tape	Specify the local tape device and the position on the tape where the Solaris Flash archive is located.
Local device	Specify the local device, the path to the Solaris Flash archive, and the type of file system on which the Solaris Flash archive is located.

   c. **On the Flash Archive Selection screen, you can install an additional Solaris Flash archive by selecting New. If you do not want to install additional archives, press Continue to complete the installation.**

When the Solaris `suninstall` program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories.

- `/var/sadm/system/logs`
- `/var/sadm/install/logs`

5. **If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to "Solaris `suninstall` Program Post-Upgrade Task" on page 201.**

6. **(Optional) To install additional software, refer to** *System Administration Guide: Basic Administration***.**

x86: Performing an Installation or Upgrade With the Solaris `suninstall` Program

You can either install or upgrade the Solaris operating environment on your x86 system by using the Solaris `suninstall` program on the Solaris Software 1 of 2 CD. You cannot install any additional software with the Solaris `suninstall` program.

Ensure that you have the following CDs:

- Solaris Software 1 of 2 *x86 Platform Edition* CD.
- Solaris Software 2 of 2 *x86 Platform Edition* CD – The installation program prompts you for this CD if necessary.
- Solaris *x86 Platform Edition* Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

x86: Performing a suninstall Installation Task Map

TABLE 18–2 x86: Performing an Interactive Installation Task Map

Task	Description	For Instructions
Gather the necessary information.	Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.	Chapter 6
(Upgrade only) Prepare to upgrade the system.	Back up the system.	*System Administration Guide: Devices and File Systems*
(Optional) Preconfigure system configuration information.	You can use the `sysidcfg` file or the name service to preconfigure installation information (for example, `locale`) for a system so the installation program does not prompt you to supply the information during the installation.	Chapter 7
(Optional) Set up the system to install from the network.	To install a system from a remote Solaris Software *x86 Platform Edition* CD image, you need to set up the system to boot and install from an install server or boot server.	Chapter 15
Install or upgrade.	Boot the system and follow the prompts to install or upgrade the Solaris software.	"x86: To Perform an Installation or Upgrade With the Solaris `suninstall` Program" on page 196
(Upgrade only) Perform post-upgrade task.	Correct any local modifications that were lost during the upgrade.	"Solaris `suninstall` Program Post-Upgrade Task" on page 201

▼ x86: To Perform an Installation or Upgrade With the Solaris suninstall Program

1. **Decide if you want to install the software by using the CD-ROM drive or by using a net image.**
   - If you're using a CD-ROM drive, insert the Solaris Software 1 of 2 *x86 Platform Edition* CD.
   - If you're using a net image, change directories to where the installation media is located. You might need to check with your network administrator for the location. The following command is an example.

     ```
     % cd /net/install-svr/export/Solaris_9/x86
     ```

2. **Decide how to boot the system.**

- If you boot from the Solaris Installation CD, insert the CD. Your system's BIOS must support booting from a CD.
- If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the system's BIOS setup tool or the network adapter's configuration setup tool.
- If you boot from a diskette, insert the Solaris Device Configuration Assistant *x86 Platform Edition* diskette into the system's diskette drive.

---

x86 only – You can copy the Device Configuration Assistant software to a diskette from the Solaris *x86 Platform Edition* DVD or Solaris Software 2 of 2 *x86 Platform Edition* CD by using the procedure in Appendix I.

---

3. **Boot the system by shutting it down and then turning it off and on.**

   A memory test and hardware detection are executed. The screen refreshes.

   - If you are using the Solaris Software 1 of 2 *x86 Platform Edition* CD and Solaris Software 2 of 2 *x86 Platform Edition* CDs, the following message appears.

     ```
     SunOS - x86 Platform Edition   Primary Boot Subsystem, vsn 2.0
     ```

     Then, information similar to the following test appears.

     ```
     SunOS Secondary Boot version 3.00


                  Solaris x86 Platform Edition Booting System


     Running Configuration Assistant...
     ```

   - If you are using PXE network boot to boot from the network, the following message appears.

     ```
     Solaris network boot...
     ```

     Then, information similar to the following text appears.

     ```
     SunOS Secondary Boot version 3.00


                  Solaris x86 Platform Edition Booting System


     Running Configuration Assistant...
     ```

   - If you are using the Solaris Device Configuration Assistant *x86 Platform Edition* diskette, the following appears.

     ```
     Solaris Boot Sector                          Version 1
     ```

Then, information similar to the following text appears.

```
Solaris for x86 - FCS DCB                    Version 1.242




                       loading /solaris/boot.bin
```

The screen refreshes and information similar to the following text appears.

```
SunOS Secondary Boot version 3.00


             Solaris x86 Platform Edition Booting System


Running Configuration Assistant...
```

4. **When the Solaris Device Configuration Assistant screen appears, press F2_Continue.**

   The Bus Enumeration screen appears with the message:

   ```
   Determining bus types and gathering hardware configuration data ...
   ```

   The Scanning Devices screen appears. System devices are scanned. When scanning is complete, the Identified Devices screen appears.

5. **Decide if you need to make any changes.**

   - Press F2_Continue to make no changes.
   - Select changes and press F4.

   The Loading screen appears and contains messages about drivers that are loaded to boot your system. After a few seconds, the Boot Solaris screen appears.

6. **On the Boot Solaris screen, select CD, Net, or Disk and press F2_Continue.**

   The Solaris `suninstall` program checks the default boot disk for the requirements to install or upgrade the system.

   Note – If you want to preserve an existing Service partition on your system, choose the appropriate option from the Create Solaris `fdisk` Partition screen. For more information on preserving the Service partition, see "x86: Change in Default Boot-Disk Partition Layout" on page 48.

   If you restored the Service partition on your system before you started the installation, you might not be able to upgrade to the Solaris operating environment.

   ---

7. **If you are prompted, answer the system configuration questions.**
   - If you preconfigured all of the system configuration information, proceed to Step 8.
   - If you did not preconfigure the system configuration information, use the "Worksheet for Installation" on page 52 or the "Worksheet for Upgrading" on page 58 to help you answer the system configuration questions.

   After a few seconds, the Solaris Installation Program screen appears.

8. **Press F2_Continue.**

   The kdmconfig – Introduction screen appears.

9. **Press F2_Continue.**

   The kdmconfig - View and Edit Window System Configuration screen appears.

10. **(Optional) Examine the configuration information on the kdmconfig – View and Edit Window System Configuration screen and make any changes you need.**

11. **When you are finished, select No changes needed – Test/Save and Exit, and press F2_Continue.**

    The kdmconfig Window System Configuration Test screen appears.

12. **Press F2_Continue.**

    The screen refreshes and the kdmconfig Window System Configuration Test palette and pattern screen appears.

13. **Move the pointer and examine the colors that are shown on the palette to ensure that they are displayed accurately.**
    - If the colors do not display accurately, click No. If possible, press any key on the keyboard, or wait until `kdmconfig` exits the kdmconfig Window System Configuration Test screen automatically. Repeat Step 10 through Step 13 until the colors are displayed accurately and you can move the pointer as expected.
    - If the colors display accurately, click Yes.

    The Solaris `suninstall` program begins.

14. **If you are prompted, answer any remaining system configuration questions.**
    - If you preconfigured all of the system configuration information, the Solaris `suninstall` program does not prompt you to enter any configuration information.
    - If you did not preconfigure the system configuration information, use the "Worksheet for Installation" on page 52 or the "Worksheet for Upgrading" on page 58 to help you answer the system configuration questions.

15. **Decide if you want to reboot the system automatically and if you want to automatically eject the disc.**

Note – If you are using PXE from a network to boot, select manual reboot. You must ensure that the system does not boot from the network when it reboots. To disable network boot, during the reboot use the system's BIOS setup tool or network adapter's configuration setup tool.

16. **Follow the instructions on the screen to install the Solaris software on the system.**

    - If you are *not* installing a Solaris Flash archive, complete the installation. If you are upgrading or need to install additional software, proceed to Step 17.

      When the Solaris `suninstall` program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

      After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories.

      - `/var/sadm/system/logs`
      - `/var/sadm/install/logs`

    - If you are installing a Solaris Flash archive, continue.

    a. **On the Flash Archive Retrieval Method screen, select the location of the Solaris Flash archive.**

    The Solaris `suninstall` program installation prompts you to proceed, depending on the media you selected.

    b. **Type the information that you are prompted to enter.**

Media Selected	Prompt
HTTP	Specify the URL and proxy information that is needed to access the Solaris Flash archive.
FTP	Specify the FTP server and the path to the Solaris Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.
Network File System	Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.
Local file	Specify the path to the local file system where the Solaris Flash archive is located.
Local tape	Specify the local tape device and the position on the tape where the Solaris Flash archive is located.

Media Selected	Prompt
Local device	Specify the local device, the path to the Solaris Flash archive, and the type of file system on which the Solaris Flash archive is located.

c. **On the Flash Archive Selection screen, you can install an additional Solaris Flash archive by selecting New. If you do not want to install additional archives, press Continue to complete the installation.**

When the Solaris `suninstall` program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

- `/var/sadm/system/logs`
- `/var/sadm/install/logs`

17. **If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to "Solaris `suninstall` Program Post-Upgrade Task" on page 201.**

18. **(Optional) To install additional software, refer to** *System Administration Guide: Basic Administration*.

Solaris `suninstall` Program Post-Upgrade Task

When you upgrade, the Solaris `suninstall` program merges local software modifications of the existing system with the new Solaris software. However, in some situations, merging is not possible. After you finish upgrading a system, you might need to correct some local modifications that were not preserved.

Caution – Your system might not boot if you do not correct the local modifications that were not preserved.

▼ To Correct Local Modifications After Upgrading

1. **Review the contents of the following file to determine whether you need to correct local modifications that the Solaris `suninstall` program could not preserve.**

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

2. **Correct any local modifications that were not preserved.**

3. **Reboot the system.**

   ```
   # reboot
   ```

CHAPTER **19**

Solaris Flash Installation Feature (Topics)

This section provides instructions for creating Solaris Flash archives and using Solaris Flash archives to install the Solaris operating environment on multiple systems.

Chapter 20	Provides overview and planning information on creating and installing Solaris Flash archives.
Chapter 21	Provides step-by-step instructions for creating Solaris Flash archives.
Chapter 22	Provides references to instructions for using Solaris Flash archives to install systems. Also provides instructions for administering archives.
Chapter 23	Describes syntax and options for the `flar create` command. Describes keywords for Solaris Flash archive information.

CHAPTER **20**

Solaris Flash (Overview and Planning)

This chapter provides an introduction to the Solaris Flash installation feature. Also included in the chapter is information necessary for planning a Solaris Flash installation in your environment.

- "Solaris Flash Introduction" on page 205
- "Planning Your Solaris Flash Installation" on page 210

Solaris Flash Introduction

The Solaris Flash installation feature enables you to use a single reference installation of the Solaris operating environment on a system, which is called the master system. Then, you can replicate that installation on a number of systems, which are called clone systems. You can replicate clone systems with a Solaris Flash initial installation that overwrites all files on the system or with a Solaris Flash update that only includes the differences from two system images. A differential update changes only the files that are specified and is restricted to systems that contain software consistent with the old master image.

Installing Clone Systems With an Initial Installation

You can install a master system with a Solaris Flash archive for an initial installation by using any installation method: Solaris Web Start, Solaris `suninstall` program, custom JumpStart, Solaris Live Upgrade or WAN boot. All files are overwritten. The Solaris Flash installation is a five–part process.

1. Install the master system. You select a system and use any of the Solaris installation methods to install the Solaris operating environment and any other software.

2. (Optional) Prepare customization scripts to reconfigure or customize the clone system before or after installation.
3. Create the Solaris Flash archive. The Solaris Flash archive contains a copy of all of the files on the master system, unless you excluded some nonessential files.
4. Install the Solaris Flash archive on clone systems. The master system and the clone system must have the same kernel architecture.

   When you install the Solaris Flash archive on a system, all of the files in the archive are copied to that system. The newly installed system now has the same installation configuration as the original master system, thus the system is called a clone system. Some customization is possible with the use of scripts.
5. (Optional) Save a copy of the master image. If you plan to create a differential archive, the master image must be available and identical to the image installed on the clone systems.

For step-by-step instructions, see "Installing the Master System" on page 220.

Figure 20–1 shows an installation of clone systems with an initial installation. All files are overwritten.

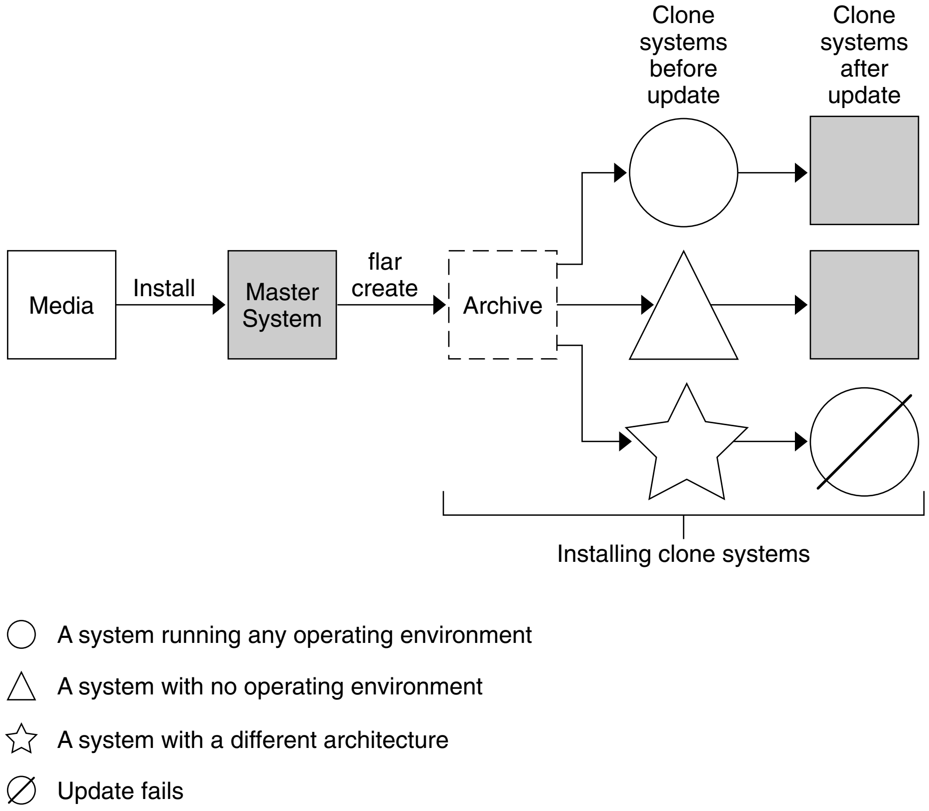

Clone systems before update
Clone systems after update
Media
Install
Master System
flar create
Archive
Installing clone systems
A system running any operating environment
A system with no operating environment
A system with a different architecture
Update fails

FIGURE 20–1 Solaris Flash Initial Installation

Updating Clone Systems With a Solaris Flash Differential Archive

If you have a clone system and want to update it, you can create a differential archive that contains only the differences between two images, the unchanged master image and an updated master image. When you update a clone system with a differential archive, only the files that are in the differential archive are changed. You can choose to install a Solaris Flash differential archive with the custom JumpStart installation method or Solaris Live Upgrade. An update is a five-part process.

1. Prepare the master system with changes. Before changes are made, the master system should be running a duplicate of the original archive.

Note – If the master system is not running a duplicate of the original archive, the differences between the two system images might result in a large differential archive. Consequently, installing the differential archive could be time consuming. Use an initial installation with a full archive in this case.

2. (Optional) Prepare customization scripts to reconfigure or customize the clone system before or after installation.
3. Mount the directory of a copy of the saved-unchanged master image. This second image is to be used to compare the two system images. Access the image by the following methods.
   - Mounted from a Solaris Live Upgrade boot environment
   - Mounted from a clone system over NFS
   - Restored from backup using the `ufsrestore` command
4. Create the differential archive with the `-A` option of the `flar create` command.
5. Install the differential archive on clone systems with custom JumpStart. Or, you can use Solaris Live Upgrade to install the differential archive on an inactive boot environment.

Figure 20–2 shows the creation and installation of a differential archive. A master image is updated with some modifications. These modifications could be as simple as the addition, reconfiguration, or deletion of a few files, or as complex as propagating patches. The updated master image is compared to the unchanged master image. The differences between the two images become the differential archive. The archive can be used to update other clone systems that are currently using the unchanged master image. If the clone system has already been modified or is not running the unchanged master image, the update fails. If you have many changes to make on the clone systems, you can do an initial installation at any time.

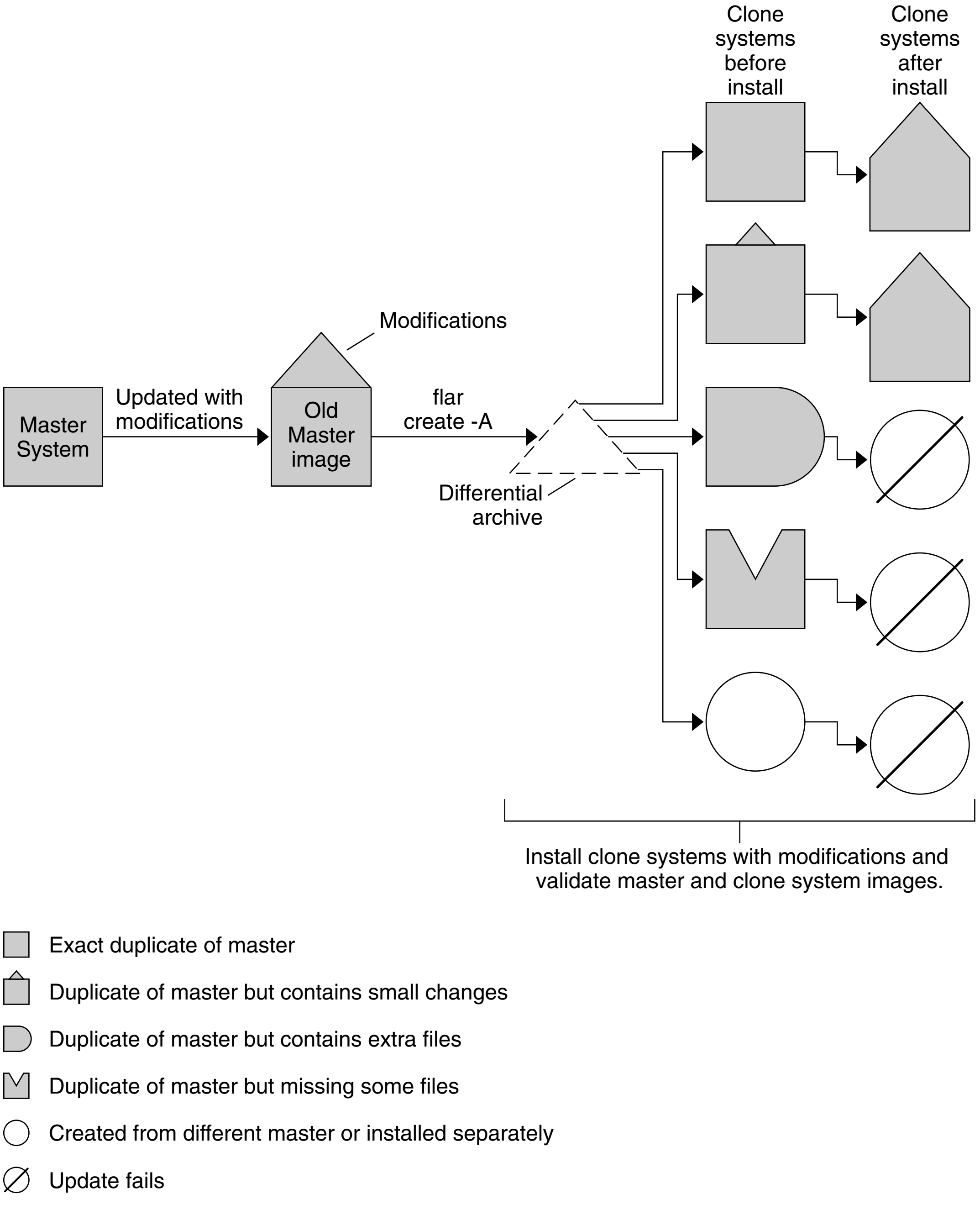

Clone systems before install
Clone systems after install
Modifications
Master System
Updated with modifications
Old Master image
flar create -A
Differential archive
Install clone systems with modifications and validate master and clone system images.
Exact duplicate of master
Duplicate of master but contains small changes
Duplicate of master but contains extra files
Duplicate of master but missing some files
Created from different master or installed separately
Update fails

FIGURE 20–2 Solaris Flash Update

Planning Your Solaris Flash Installation

Before you create and install a Solaris Flash archive, you must make some decisions about how you want to install the Solaris operating environment on your systems. The first time you install a system, you need to install with a full archive that is an initial installation. After a system has been installed with an archive, the system can be updated with a differential archive. The differential archive installs only the differences between two archives.

Designing an Initial Installation of the Master System

The first task in the Solaris Flash installation process is to install a system, the master system, with the configuration that you want each of the clone systems to have. You can use any of the Solaris installation methods to install an archive on the master system. The installation can be a subset or a complete installation of the Solaris operating environment. After you complete the installation, you can add or remove software or modify any configuration files.

The master system and the clone systems must have the same kernel architectures. For example, you can only use an archive that was created from a master system that has a `sun4u` architecture to install clones with a `sun4u` architecture.

You must install the master system with the exact configuration that you want on each of the clone systems. The decisions that you make when you design the installation of the master system depend on the following:

- The software that you want to install on the clone systems
- Peripheral devices that are connected to the master system and the clone systems
- The architecture of the master system and the clone systems

The software on the system can then be customized. See “Customizing an Archive’s Files and Directories” on page 214. The master system can also be modified and set up for an update of a clone system. See “Planning the Creation of a Solaris Flash Archive” on page 212.

Customizing the Solaris Installation on the Master System

After you install the Solaris operating environment on the master system by using any of the Solaris installation methods, you can add or delete software and modify system configuration information as necessary.

- Delete software. You can remove software that you determine is not necessary to install on the clone systems. To see a list of software that is installed on the master system, use the Product Registry. For detailed instructions, refer to *System Administration Guide: Basic Administration*.
- Add software. You can install software that is included in the Solaris release. You can also add software that is not delivered as part of the Solaris operating environment. All of the software that you install on the master system is included in the Solaris Flash archive and is installed on the clone systems.
- Modify configuration files. You can alter configuration files on the master system. For example, you can modify the `/etc/inet/inetd.conf` file to restrict the daemons that the system runs. All of the modifications that you make are saved as part of the Solaris Flash archive and are installed on the clone systems.

Further customization can be done when creating the archive. For example, you can exclude large data files that you might not want in the archive. For an overview, see "Customizing an Archive's Files and Directories" on page 214.

To update a clone system, you can create a differential archive. See "Planning the Creation of a Solaris Flash Archive" on page 212.

Creating Archives for SPARC and x86 Systems

If you want to use the Solaris Flash installation method to install the Solaris software on both SPARC and x86 systems, you must create a separate Solaris Flash archive for each platform. Use the Solaris Flash archive that was created from the SPARC master system to install SPARC systems. Use the Solaris Flash archive that was created from the x86 master system to install x86 systems.

SPARC: Supporting Peripheral Devices Not Found on the Master System

Choosing the drivers to install on the master system depends on the type of peripheral devices attached to both the master system and the clone system and depends on the type of software group installed. If the clone systems have different peripheral devices than the master system, you need to install the appropriate drivers on the master system before you create the archive.

Note – When you install Solaris software, some hardware-specific drivers are installed only if their supported hardware is found on the system.

If you install the master system with the Core, End User, Developer, or Entire Software Group, the master system supports only the peripheral devices that are attached to the master system at the time of installation. You can install support for peripheral devices on the master system even though the master system does not have the devices by installing the master system with the Entire Plus OEM Software Group. All drivers are installed regardless of the hardware present on the system. Or, you can install selected packages that support the peripherals needed.

If you install any software group except the Entire Plus OEM Software Group on your master system, you might not have all the drivers that you need to correctly install your clone systems. For example, if you install the Entire Software Group on a master system that has a GX CG6 frame buffer, only the GX CG6 frame buffer driver is installed. This situation is not a problem if all the clone systems you want to install have the GX CG6 frame buffer or no frame buffer.

If you want to install the archive on clone systems that have other frame buffers installed, such as the Elite 3D, then the clone system does not have frame buffer support. Therefore, you must create an archive that includes all frame buffers by installing the Entire Plus OEM Software Group or installing the frame buffer drivers you want by adding the appropriate frame buffer driver packages to the master system.

You can install support for peripherals on clone systems that are different from the master system in one of the following ways.

- Install the Entire Plus OEM Software Group – The Entire Plus OEM Software Group is the largest Software Group available. This group contains every package that is found in the Solaris operating environment. If you install this group on the master system, the master system includes all of the drivers that are included with the Solaris release. A Solaris Flash archive that is created from a master system that you install with the Entire Plus OEM Software Group works on any clone system that has peripheral devices supported by the installed release of the Solaris operating environment.

  Installing master systems with the Entire Plus OEM Software Group guarantees compatibility with other peripheral configurations. However, the Entire Plus OEM Software Group requires over 1 Gbyte of disk space. The clone systems might not have the space that is required to install the Entire Plus OEM Software Group.
- Install selected packages – When you install the master system, you can install only the packages that you need for the master system and the clone systems. By selecting specific packages, you can install only support for the peripherals that you know exist on the master system or clone systems.

Planning the Creation of a Solaris Flash Archive

You can create an archive from the master system or create a differential archive from two system images. The differential archive installs only the differences between the two images.

Planning to Create the Solaris Flash Archive for an Initial Installation

After you install the master system, the next task in the Solaris Flash installation process is to create a Solaris Flash archive. Files on the master system are copied to a Solaris Flash archive along with various pieces of identification information. You can create a Solaris Flash archive while the master system is running in multiuser mode or single-user mode. You can also create a Solaris Flash archive after you boot from one of the following:

- Solaris DVD
- Solaris Software 1 of 2 CD
- An image of the Solaris Software and the Solaris Languages CDs

Planning to Create the Solaris Flash Differential Archive for an Update

If you have a clone system and want to update it, you can create a differential archive that contains only the differences between two images, the unchanged master image and an updated master image. One image can be your system that is running the original software that was installed on the clone, or you need to access a saved copy of the unchanged master image. This image is to be updated with the desired changes. Another image is to be accessed and used for comparison. The differences between these two images is the differential archive. You can install a Solaris Flash differential archive with the custom JumpStart installation method. Or, you can use Solaris Live Upgrade to install a differential archive on an inactive boot environment. After you update a clone system with a differential archive, only the files that are in the differential archive are changed on the clone system. Scripts can be used to customize the archive before or after installation, which is especially helpful for reconfiguration.

An unchanged master image should be saved after the initial installation so it could be accessed by any of the following methods.

- A Solaris Live Upgrade boot environment, mounted on some directory that uses the `lumount` command. For a description of a Solaris Live Upgrade boot environment, see Chapter 33.
- A clone system that is mounted over NFS with root permissions.
- A system backup that can be restored with the `ufsdump` command.

For step-by-step instructions, see "To Create a Solaris Flash Differential Archive With an Updated Master Image" on page 231.

Customizing an Archive's Files and Directories

When you create a Solaris Flash archive, some files and directories that are to be copied from the master system can be excluded. If you have excluded a directory, you can also restore specified files or subdirectories under that directory. For example, you could create an archive that excludes all files and directories in `/a/aa/bb/c`. The content of the `bb` subdirectory could be included. The only content then would be in the `bb` subdirectory.

Caution – Use the `flar create` file-exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.

The following table lists the `flar create` command options that can exclude files and directories and restore files and subdirectories.

How Specified?	Options That Exclude	Options That Include
Specify the name of the directory or file	`-x` *exclude_dir/filename*	`-y` *include_dir/filename*
Use a file that contains a list	`-X` *list_filename* `-z` *list_filename*	`-f` *list_filename* `-z` *list_filename*

For descriptions of these options, see Table 23–7.

For examples of customizing an archive, see "Examples—Creating an Archive for an Initial Install" on page 227.

Customizing an Archive With Scripts

After the software is installed on the master system, special scripts can be run during creation, installation, postinstallation and first reboot. These scripts enable you to do the following:

- Configure applications on clone systems. You can use a custom JumpStart script for some uncomplicated configurations. For more complicated configurations, special configuration file processing might be necessary on the master system or before or after installation on the clone system.
- Protect local customizations on clone systems. Local preinstallation and postinstallation scripts reside on the clone and protect local customizations from being overwritten by the Solaris Flash software.

- Identify nonclonable, host-dependent data that enables you to make the archive host independent. Host independence is enabled by modifying such data or excluding it from the archive. An example of host-dependent data is a log file.
- Validate software integrity in the archive during creation.
- Validate the installation on the clone system.

Guidelines for Creating a Custom Script

When creating a scripts other than the reboot script, following these guidelines to assure the script does not corrupt the operating environment or otherwise disrupt the system. These guidelines enable the use of Solaris Live Upgrade, which creates a new boot environment for installation of the operating environment. The new boot environment can be installed with an archive while the current system is running.

Note – These guidelines are not for reboot scripts that are allowed to run daemons or make other types of modification to the root (/) file system.

- Scripts must not affect the currently running system. The currently running operating environment might not be the one running when the Solaris Flash archive is installed.
- Scripts must not start or stop any daemon processes.
- Scripts must not depend on the output of commands such as `ps`, `truss`, or `uname`, which are dependent on the operating environment. These commands report information about the currently running system.
- Scripts must not send any signals or otherwise affect any currently running processes.
- Scripts can use standard UNIX commands that facilitate shell scripting such as `expr`, `cp`, and `ls`.

For an overview of Solaris Live Upgrade, see Chapter 33.

Solaris Flash Archive Sections

Solaris Flash archives contain the following sections. Some sections can be used by you to identify and customize the archive and view status information on the installation. For a further description of each section, see Chapter 23.

TABLE 20–1 Flash Archive Sections

Section Name	Informational Only	Description
Archive cookie	X	The first section contains a `cookie` that identifies the file as a Solaris Flash archive.
Archive identification		The second section contains keywords with values that provide identification information about the archive. Some identification information is supplied by the archive software. Other specific identification information can be added by you by using options to the `flar create` command.
User-defined		This section follows the identification section. You can define and insert these sections to customize the archive. The Solaris Flash archive does not process any sections that you insert. For example, a section could contain a description of the archive or perhaps a script to check the integrity of an application.
Manifest	X	This section is produced for a Solaris Flash differential archive and used for validating a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. This section is informational only, lists the files in an internal format, and cannot be used for scripting.
Predeployment, Postdeployment, Reboot	X	This section contains internal information that the flash software uses before and after installing an operating environment image. Any scripts that you have provided are included in this section.
Summary		This section contains messages about the archive creation. The section also records the activities of predeployment and postdeployment scripts. You can view the success of the installation in this section by writing a script to send output to this section.
Archive files	X	The archive files section contains the files that have been gathered from the master system.

When to Create the Archive for an Initial Installation

Create the archive when the system is in as static a state as possible.

Create the archive after software is installed on the master system and before software is configured. For example, Solaris Volume Manager stores metainformation outside the file system. If you have an encapsulated and mirrored boot disk, the creation of the archive is not possible. Because archive creation does not access the metainformation outside the file system, the archive must be created before configuration.

Where to Store the Solaris Flash Archive

After you create the Solaris Flash archive, you can save the archive on the hard disk of the master system or on a tape. After you save the archive, you can copy it to any file system or media that you choose.

- Network File System (NFS) server
- HTTP or HTTPS server
- FTP server
- Tape
- CD, DVD
- Diskette
- Local drive of clone system that you want to install

Compressing the Archive

When you create the Solaris Flash archive, you can specify that the archive be saved as a compressed file by using the `compress(1)` utility. An archive that is compressed requires less disk storage space and creates less congestion when you install the archive over a network.

Planning the Installation of Solaris Flash Archives

The final task in the Solaris Flash installation process is to install Solaris Flash archives on clone systems.

You can use any of the Solaris installation methods to install Solaris Flash archives on clone systems.

The Solaris Web Start program on the Solaris DVD or Solaris Installation CD enables you to install Solaris Flash archives that are stored on the following:

- Disc (DVD or CD)
- NFS server
- HTTP server
- FTP server
- Local tape

For installation instructions, see "SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program" on page 176 or "x86: Performing an Installation or Upgrade With the Solaris Web Start Program" on page 181.

The Solaris `suninstall` program on the Solaris Software 1 of 2 CD enables you to install Solaris Flash archives that are stored on the following:

- NFS server
- HTTP server

- FTP server
- Local file
- Local tape
- Local device, including CD

For installation instructions, see "SPARC: Performing an Installation or Upgrade With the Solaris `suninstall` Program" on page 192 or "x86: Performing an Installation or Upgrade With the Solaris `suninstall` Program" on page 195.

The custom JumpStart installation program enables you to install Solaris Flash archives that are stored on the following:

- NFS server
- HTTP or HTTPS server
- FTP server
- Local tape
- Local device, including DVD or CD
- Local file

For installation instructions, see "To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation" on page 317.

Solaris Live Upgrade enables you to install Solaris Flash archives that are stored on the following:

- HTTP server
- FTP server
- NFS server
- Local file
- Local tape
- Local device, including DVD or CD

For installation instructions, see "Installing Solaris Flash Archives on a Boot Environment" on page 456.

CHAPTER 21

Creating Solaris Flash Archives (Tasks)

This chapter provides the procedures for creating a Solaris Flash archive. These procedures include installing a master system and then creating a Solaris Flash archive from that master system. You can also create a differential archive if you have previously installed an archive on a clone system. When the differential archive is created, two images are compared: the unchanged master image and an updated image. The differential archive installs only the differences between the two images. Also, procedures to create scripts are provided to reconfigure or otherwise customize the archive.

- "Task Map: Creating Solaris Flash Archives " on page 219
- "Installing the Master System" on page 220
- "Creating Customization Scripts" on page 221
- "Creating a Solaris Flash Archive" on page 226

Task Map: Creating Solaris Flash Archives

TABLE 21–1 Task Map: Creating a Solaris Flash Archive to Install With an Initial Installation

Task	Description	For Instructions
Install your chosen software configuration on the master system	Determine the configuration that meets your needs and use any of the Solaris installation methods to install the master system.	"To Install the Master System for an Initial Installation" on page 221

TABLE 21–1 Task Map: Creating a Solaris Flash Archive to Install With an Initial Installation *(Continued)*

Task	Description	For Instructions
(Optional) Create customization scripts	Determine if you need to create scripts to do the following: ■ Customize or reconfigure the archive ■ Protect local changes on clone systems	"Creating Customization Scripts" on page 221
Create the Solaris Flash archive	Use the `flar create` command to create an archive.	"To Create a Solaris Flash Archive for an Initial Installation" on page 226
(Optional) Save a copy of the archive	Keep a copy of the archive for future comparison to update a clone system with a differential archive.	"To Create a Solaris Flash Archive for an Initial Installation" on page 226

TABLE 21–2 Task Map: Creating a Solaris Flash Archive to Update a Clone System

Task	Description	For Instructions
Prepare master image	Make changes to the unchanged master image such as adding or deleting packages or installing patches.	"Installing the Master System" on page 220
(Optional) Create customization scripts	Determine if you need to create scripts to do the following: ■ Customize or reconfigure the archive ■ Protect local changes on clone systems	"Creating Customization Scripts" on page 221
Create the Solaris Flash differential archive	1. Mount the unchanged master image. 2. Use the `flar create` command to compare the two images and create the differential archive.	"To Create a Solaris Flash Differential Archive With an Updated Master Image" on page 231

Installing the Master System

You install the master system with the software configuration that you want other systems to have. You can install clone systems with an initial installation that overwrites all files on the system or with an update that only includes the differences between two images. For an initial installation, use any of the Solaris installation methods to install the Solaris operating environment on the master system.

If you have previously installed an archive on a clone system, you can update that system with changes by using a differential archive. The changes are made to the original image such as installing patches, or adding and removing packages. The differential archive overwrites only the files specified in the archive. For the procedure for updating the original master image and creating a differential archive, see "To Create a Solaris Flash Differential Archive With an Updated Master Image" on page 231.

▼ To Install the Master System for an Initial Installation

1. **Identify the system configuration that you want to install.**

2. **With the use of the Solaris installation methods, install the Solaris operating environment on the master system. For a discussion of the different installation methods, refer to Chapter 3.**

3. **Customize your Solaris installation in any of the following ways:**

   - Delete software.
   - Add software.
   - Modify configuration files.
   - Add support for peripheral devices on the clone system.

   You can create custom scripts or use the `flar create` command to create the archive.

   - To create custom scripts, see "Creating Customization Scripts" on page 221.
   - To create the archive, see "Creating a Solaris Flash Archive" on page 226.

Creating Customization Scripts

Scripts can customize the archive. Use these scripts for the following purposes:

- A precreation script validates the archive at creation time and prepares the archive for later customization, especially differential archives. This script also can create a user-defined section in the archive.
- A predeployment script validates the archive during installation and prepares the archive for later customization.
- A postdeployment script reconfigures a new system image on a clone system.
- A reboot script processes a final reconfiguration after the system is rebooted.

For guidelines on creating scripts, see "Guidelines for Creating a Custom Script" on page 215.

▼ To Create a Precreation Script

This script runs during archive creation. The script has various uses.

- Validates the contents and the integrity of the software. The script fails the archive creation if the integrity is broken.
- Prepares products for later customization on clone system.
- Registers other installation scripts dynamically during archive creation.
- Adds a message to the flash-creation summary file. The message must be short and record only that scripts were started and finished and the results. You can view the results in the summary section.

1. **Create the precreation script. Follow the guidelines that are described in "Guidelines for Creating a Custom Script" on page 215.**
2. **Store the script in the `/etc/flash/precreation` directory.**

EXAMPLE 21–1 Excerpts From a Precreation Script

The following examples are excerpts from a precreation script.

- To log the start time in the summary section, use the following example:

  ```
  echo "MyApp precreation script started">> $FLASHDIR/summary
  ```

- To check the software integrity, use the `flcheck` command. This command cannot be used at the command line. The syntax for this command is as follows:

  ```
  flcheck software component files and directories ...| -
  ```

  For example, to validate the files and directories, use the following example:

  ```
  flcheck software component files and directories
  If Not in selection - refuse creation

  echo "Myapp Integrity Damage">>$FLASHDIR/summary
  ```

  Or, to keep new files and directories that are unexpected and not fail the archive creation, use the following example:

  ```
  flcheck software component files and directories
  If Not in selection include by force
  flinclude software component
  ```

- To register deployment scripts and data, use the following example:
  - Copy the script to the following directory:

    ```
    cp predeployment script /etc/flash/predeployment
    ```

  - Or, to register the script dynamically during archive creation, copy the script to the following directory.

    ```
    cp predeployment script $FLASHDIR/predeployment
    ```

EXAMPLE 21–1 Excerpts From a Precreation Script *(Continued)*

- To see application specific data in a user-defined section, use the following example:

  ```
  cp custom section $FLASHDIR/custom_sections/MyApp
  ```

- To log the success of the installation in the summary section, use the following example:

```
echo "product one flash preparation started." >>$FLASH_DIR/summary
...
echo "product one flash preparation finished successfully">>$FLASH_DIR/summary
```

EXAMPLE 21–2 Precreation Script

```
#!/bin/sh
echo "Test precreation script started" >> $FLASH_DIR/summary
cat /opt/TestApp/critical_file_list | flcheck -
if [ $? != 0 ]; then
    echo "Test precreation script failure" >> $FLASH_DIR/summary
    exit 1
fi
echo "Test precreation script started" >> $FLASH_DIR/summary
/opt/TestApplication/license_cloning
    $FLASH_DIR/predeployment/.TestApplicationLicenceTransfer \
    $FLASH_DIR/custom_sections/TestApplicationLicenceCounter
echo "Test precreation script finished" >> $FLASH_DIR/summary
exit 0
```

Using a Precreation Script to Create a User-Defined Archive Section

A precreation script can create a user-defined section in the archive to provide specific application information. This section is intended for archive maintenance. The script must be put in the `$FLASH_DIR/sections` directory. The Solaris Flash archive does not process a user-defined section. For example, a section could contain a description of the archive or perhaps a script to check the integrity of an application.

A user-defined section requires the following format.

- Must be line oriented
- Must terminate with newline (ASCII 0x0a) characters
- Can have unlimited length of individual lines
- Encode binary data by using base64 or a similar algorithm

▼ To Create a Predeployment Script

This script is run before the installation of the archive. If the script is meant to validate the archive, it is kept in the archive. If the script is meant to preserve local configurations on the clone system, it is kept on the clone system. This script also can analyze and collect local data necessary for later customization. For example, client-specific information can be saved before being overwitten by files about to be extracted. This information can then be used in the final stage after extraction.

1. **Create the predeployment script. Follow the guidelines that are described in "Guidelines for Creating a Custom Script" on page 215.**

2. **Store the script in one of the following directories.**
   - For archive validation, store in the /etc/flash/predeployment directory.
   - If you are referencing with a precreation script, store in the $FLASH_DIR/preinstall directory.
   - If you are preserving configurations on a clone system, provide the path to the script that is stored on the clone system with the local_customization keyword in the JumpStart profile.

EXAMPLE 21–3 Predeployment Script

```
#!/bin/sh
$FLASH_DIR/TestApplication/check_hardware
if [ $? != 0 ]; then
    echo Unsupported hardware
    exit 1
fi
$FLASH_DIR/TestApplication/check_licence_key
if [ $? != 0 ]; then
    echo No license for this host
    exit 1
fi
$FLASH_DIR/TestApplication/deploy_license_key \
    $FLASH_DIR/TestApplication/.TestApplicationLicenceTransfer
$FLASH_DIR/TestApplication/save_data_files $FLASH_DIR/flash

exit 0
```

▼ To Create a Postdeployment Script

This script is kept in the archive or stored in a local directory on the clone system and runs after installation. The script reconfigures a new system image on a clone system. If the script is stored in the archive, the changes affect all the clone systems. If the script is stored in a local directory on the clone system, the changes affect only the clone system. For example, client-specific information that is saved by a predeployment script can be applied to the clone environment, completing the installation.

Postdeployment scripts can also be used to clean up files after the archive is installed. For example, log files such as those files in /var/adm can be cleaned out.

Note – Not all log files need a script for cleanup. Log files in /var/tmp can be excluded when the archive is created.

1. **Create the postdeployment script. Follow the guidelines described in "Guidelines for Creating a Custom Script" on page 215.**
2. **Store the script in one of the following directories.**
   - To affect all clone systems, store the script in the /etc/flash/postdeployment directory.
   - To affect only a local clone system, provide the path to the script that is stored on the clone system with the local_customization keyword in the JumpStart profile.

EXAMPLE 21–4 Postdeployment Script

```
#!/bin/sh
$FLASH_DIR/TestApplication/clone_reconfiguration
$FLASH_DIR/TestApplication/restore_data $FLASH_DIR/flash
```

▼ To Create a Reboot Script

This script is kept in the archive and runs after the system is rebooted. The script does any final configuration after system reconfiguration.

After you install the Solaris Flash archive on a clone system, some host-specific files are deleted and are re-created for the clone machine. The installation program uses the sys-unconfig(1M) command and the sysidtool(1M) programs to delete and re-create host-specific network configuration files. The files that are re-created include, for example, /etc/hosts, /etc/defaultrouter, and /etc/defaultdomain. You can use the reboot script for any final reconfiguration.

1. **Create the reboot script.**
2. **Store the script in the /etc/flash/reboot directory.**

EXAMPLE 21–5 Creating a Reboot Script

```
#!/bin/sh
$FLASH_DIR/TestApplication/finalize_license
```

Creating a Solaris Flash Archive

You can create an archive with an initial installation that overwrites all the files on the clone system, or you can create a differential archive that only overwrites the changes that are specified. For an overview of a differential archive, see "Planning to Create the Solaris Flash Differential Archive for an Update" on page 213.

▼ To Create a Solaris Flash Archive for an Initial Installation

After you install the master system, create a Solaris Flash archive to use to install other systems.

1. **Boot the master system and run it in as inactive a state as possible.**

   When possible, run the system in single-user mode. If that is not possible, shut down any applications that you want to archive and any applications that require extensive operating system resources.

   You can create a Solaris Flash archive while the master system is running in multiuser mode, single-user mode, or while booted from one of the following:

   - Solaris DVD.
   - Solaris Software 1 of 2 CD.
   - An image of the Solaris Software. If you are using CD media, the image could include the Solaris Languages CD if needed.

2. **To create the archive, use the `flar create` command.**

   ```
   # flar create -n name options path/filename
   ```

   name The name that you give the archive. The *name* you specify is the value of the `content_name` keyword.

   options For a description of options, see "`flar create`" on page 250.

   path The path to the directory in which you want to save the archive file. If you do not specify a path, `flar create` saves the archive file in the current directory.

   filename The name of the archive file.

   - If the archive creation is successful, the `flar create` command returns an exit code of 0.
   - If the archive creation fails, the `flar create` command returns a nonzero exit code.

3. **Make a copy of the archive and save it. The copy can be used in the future to update a clone system with a differential archive.**

Examples—Creating an Archive for an Initial Install

File systems can be copied exactly or can be customized by excluding some directories or files. You can achieve the same results by using different options. Use the options that best suit your environment.

The file systems in the following examples have been greatly simplified for clarification. Rather than use file system names such as `/var`, `/usr`, or `/opt`, the master system file structure for these examples is the following:

```
/aaa/bbb/ccc/ddd
/aaa/bbb/fff
/aaa/eee
/ggg
```

Caution – Use the `flar create` file exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.

EXAMPLE 21–6 Creating an Exact Duplicate Archive

In this example, the archive is named `archive1`. It is copied exactly from the master system and then compressed. The archive is an exact duplicate of the master system and stored in `archive1.flar`.

```
# flar create -n archive1 -c archive1.flar
```

To check on the file structure of the archive, type the following.

```
# flar info -l archive1.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb/fff
aaa/eee
aaa/eee
ggg
```

EXAMPLE 21–7 Creating an Archive and Excluding and Including Files and Directories

In this example, the archive is named `archive2`. It is copied from the master system but is not an exact copy. The content under the /aaa directory is excluded, but the content in `/aaa/bbb/ccc` remains.

```
# flar create -n archive2 -x /aaa -y /aaa/bbb/ccc  archive2.flar
```

To check on the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l aaa
aaa
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb
ggg
```

EXAMPLE 21–8 Creating an Archive Excluding and Including Files and Directories by Using Lists

In this example, the archive is named `archive5`. It is copied from the master system but is not an exact copy.

The `exclude` file contains the following list:

```
/aaa
```

The `include` file contains the following list:

```
/aaa/bbb/ccc
```

The content under the /aaa directory is excluded, but the content in `/aaa/bbb/ccc` remains.

```
# flar create -n archive5 -X exclude -f include  archive5.flar
```

To check on the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l archive5.flar
aaa
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb
ggg
```

EXAMPLE 21–9 Creating an Archive Excluding Files and Directories by Using a List and Restoring a Directory

You can combine options `-x`, `-y`, `-X` and `-f`. In this example, options `-X` and `-y` are combined. The archive is named `archive5`. It is copied from the master system but is not an exact copy.

The `exclude` file contains the following list:

```
/aaa
```

The `-y` option restores the `/aaa/bbb/ccc` directory. The following command produces the archive.

```
# flar create -n archive5 -X exclude -y /aaa/bbb/ccc  archive5.flar
```

To check on the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l archive5.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
ggg
```

EXAMPLE 21–10 Creating an Archive Excluding and Including Files and Directories by Using a List With the `-z` Option

In this example, the archive is named `archive3`. It is copied from the master system but is not an exact copy. The files and directories to be selected are included in `filter1` file. Within the files, the directories are marked with a minus (-) or a plus (+) to indicate which files to exclude and restore. In this example, the directory `/aaa` is excluded with a minus and the subdirectory `/aaa/bbb/ccc` is restored with a plus. The `filter1` file contains the following list.

```
- /aaa
+ /aaa/bbb/ccc
```

The following command produces the archive.

```
# flar create -n archive3 -z filter1 archive3.flar
```

To check on the file structure of the archive, type the following command. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l archive3.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
```

EXAMPLE 21–10 Creating an Archive Excluding and Including Files and Directories by Using a List With the `-z` Option *(Continued)*

```
ggg
```

EXAMPLE 21–11 Creating an Archive From an Alternate Boot Environment

In this example, the archive is named `archive4`. It is copied exactly from the master system and then compressed. The archive is an exact duplicate of the master system and stored in `archive4.flar`. The `-R` option is used to create the archive from another directory tree.

```
# flar create -n archive4 -c -R /x/yy/zz archive4.flar
```

EXAMPLE 21–12 Creating an Archive and Adding Keywords to Describe the Archive

In this example, the archive is named `archive3`. This archive is copied exactly from the master system and then compressed. Options add descriptions to the archive identification section, which can help you to identify the archive later. For information on keywords, their values, and formats, see "Solaris Flash Keywords" on page 245.

```
# flar create -n archive3 -i 20000131221409 -m pumbaa \
-e "Solaris 8 Print Server" -a "Mighty Matt" -U "Internal Finance" \
-T server archive3.flar
```

After the archive is created, you can access the identification section that contains the detailed description. An example of an identification section follows.

```
section_begin=identification
     files_archived_method=cpio
     files_compressed_method=compress
     files_archived_size=259323342
     files_unarchived_size=591238111
     creation_date=20000131221409
     creation_master=pumbaa
     content_name=Finance Print Server
     content_type=server
     content_description=Solaris 8 Print Server
     content_author=Mighty Matt
     content_architectures=sun4u,sun4m
     creation_node=pumbaa
     creation_hardware_class=sun4u
     creation_platform=SUNW,Sun-Fire
     creation_processor=sparc
     creation_release=5.9
     creation_os_name=SunOS
     creation_os_version=s81_49
     x-department=Internal Finance
```

▼ To Create a Solaris Flash Differential Archive With an Updated Master Image

Before creating a differential archive, you need two images to compare: an unchanged master image and an updated master image. One image is the unchanged master image that has been kept unchanged. This image was stored and needs to be accessed. The second image is the unchanged master image that is updated with minor changes. The root (/) file system is the default for the new image, but you can access this image if it has been stored elsewhere. After you have the two images, you can create a differential archive, which contains only the differences between the two images. The differential archive can then be installed on clones that were installed previously with the unchanged master image.

1. **Prepare the master system with changes. Before changes are made, the master system should be running a duplicate of the original archive.**

   Note – A copy of the unchanged master image must be kept protected from changes and available for mounting later.

   ---

2. **Update the unchanged master image with any of the following changes.**
   - Delete packages.
   - Add packages or patches.
   - Modify configuration files.
   - Add support for peripheral devices on the clone system.
3. **(Optional) Create custom scripts. See "Creating Customization Scripts" on page 221.**
4. **Deliver the unchanged-master image in a mount point.**
   - If the unchanged master image is stored on an inactive boot environment, retrieve by using the `lumount` command.

     ```
     # lumount BE_name mountpoint
     ```

     BE_name Specifies the boot environment name where the unchanged master image is stored

     mountpoint Specifies a root (/) file system where the image is stored

     In the following example, the inactive boot environment is named unchanged_master1. The mount point is the directory `/a` on the master system.

     ```
     # lumount  unchanged_master1 /a
     ```
   - If the image is stored on a clone, mount the clone by using NFS.

a. On the master system, share the clone's root (/) file system and give the master root permissions on the clone system.

```
# share -F nfs -o rw,root=master_system "/"
```

master_system is the name of the master system.

b. On the master system, mount the clone.

```
# mount -F nfs clone_system:/ master_dir
```

clone_system Specifies the name of the system to be mounted

master_dir Specifies the directory where the unchanged master image is stored

- If you saved the image with the `ufsdump` command, use the `ufsrestore` command to retrieve a copy. For information on how to use these commands, see "UFS Backup and Restore Commands (Reference)" in *System Administration Guide: Devices and File Systems*.

5. **Create the differential archive.**

```
# flar create -n archive_name -A unchanged_master_image_dir\
  options path/filename
```

archive_name Specifies the name that you give the archive. The *archive_name* you specify is the value of the `content_name` keyword. The name is listed in the archive identification section.

`-A` *unchanged_master_image_dir* Creates a differential archive by comparing a new system image with the image that is specified by the *unchanged_master_image_dir* argument. By default, the new system image is root (/). You can change the default with the `-R` option. *unchanged_master_image_dir* is a directory where the unchanged system image is stored or mounted through UFS, NFS, or `lumount`.

You can include and exclude some files by using the options for contents selection. For a list of options, see "`flar create`" on page 250.

options For a description of options, see "`flar create`" on page 250.

path Specifies the path to the directory in which you want to save the archive file. If you do not specify a path, `flar create` saves the archive file in the current directory.

filename Specifies the name of the archive file.

- If the differential archive creation is successful, the `flar create` command returns an exit code of 0.
- If the differential archive creation fails, the `flar create` command returns a nonzero exit code.

For procedures on installing an archive, see "To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation" on page 317.

Examples—Creating a Differential Archive

EXAMPLE 21–13 Creating a Differential Archive With the New Master Image On the Master System

In this example, the directory for unchanged master image is named `unchanged_master1`. The new master image that contains changes is the root (/) directory. The new master image is compared to the unchanged master image and the resulting differential archive is then compressed. The differential archive is stored in `diffarchive1.flar` file. The archive contains files that are to be deleted, changed, or added when installed.

```
# flar create -n diffarchive1 -A /a/unchanged_master1 -c diffarchive1.flar
```

EXAMPLE 21–14 Creating a Differential Archive With the Images Stored on an Inactive Boot Environment

In this example, the unchanged master image, `unchanged_master1`, is stored on an inactive boot environment and is accessed by mounting the boot environment. The new master image is the root (/) directory. The new master image is compared to the unchanged master and the resulting differential archive is then compressed. The archive is stored in `diffarchive4.flar`. The archive contains files that are to be deleted, changed, or added when installed.

```
# lumount unchanged_master1 /a
# flar create -n diffarchive4 -A /a -c  diffarchive4.flar
```

▼ To Create a Solaris Flash Differential Archive By Using Solaris Live Upgrade

To manage system updates, you can use Solaris Live Upgrade to copy the operating environment, which creates a new boot environment. This copy can be compared to the master system that has been updated with minor changes. The resulting Solaris Flash differential archive can then be installed on clone systems.

For more information about Solaris Live Upgrade, see Chapter 32.

1. **From the unchanged master system, create a new boot environment by using the `lucreate` command.**

   This new boot environment is an exact copy of the master system and can be used to create the differential archive.

2. **Check the status of the two boot environments.**

```
# lustatus copy_BE

boot environment   Is         Active  Active     Can        Copy
Name               Complete   Now     OnReboot   Delete     Status
------------------------------------------------------------------------
master_BE          yes        yes     yes        no         -
copy_BE            yes        no      no         yes        -
```

3. **Update the master image with any of the following changes.**

   - Delete packages.
   - Add packages or patches.
   - Modify configuration files.
   - Add support for peripheral devices on the clone system.

4. **(Optional) Create custom scripts. See "Creating Customization Scripts" on page 221.**

5. **Create the differential archive.**

   a. **Mount the newly created boot environment.**

```
# lumount BE_name /a
```

   b. **Create the differential archive by comparing the master system to the boot environment.**

```
# flarcreate -n archive_name -A new_BE_dir\
options path/filename
```

   archive_name Specifies the name that you give the archive.

   `-A` *new_BE_dir* Creates a differential archive by comparing a new system image with the image that is specified by the *new BE_dir* argument.

   options For a list of options, see "`flar create`" on page 250.

   path Specifies the path to the directory in which you want to save the archive file. If you do not specify a path, `flar create` saves the archive file in the current directory.

   filename Specifies the name of the archive file.

   c. **Unmount the new boot environment.**

```
# luumount copy_BE
```

The `flar create` command returns an exit code.

- If the creation is successful, an exit code of 0 is returned.
- If a failure occurs, a nonzero exit code is returned.

6. **Install the Solaris Flash differential archive by using a JumpStart profile.**

   The clone systems that are to be installed must be a duplicate of the original master system or the installation fails.

   The following example profile installs a differential archive, `test.diff`, on the device `c1t1d0s0`.

   ```
   JumpStart profile
   -----------------------
   install_type  flash_update
   archive_location http server /rw/test.diff
   root_device c1t1d0s0
   ```

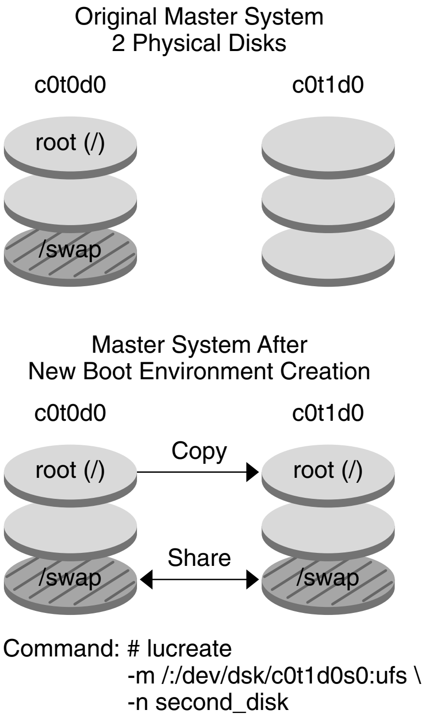
EXAMPLE 21–15 Creating a Differential Archive by Using Solaris Live Upgrade

In this example, `master_BE` is the name of the current boot environment. `copy_BE` is the name of the new boot environment. The file systems root (`/`) and `/usr` are placed on `s0` and `s3`. The `lustatus` command reports that the new boot environment copy is complete. The `SUNWman` package is added to the master system. After the master system is updated by adding the `SUNWman` package, the `flarcreate` command creates a differential archive by comparing the changed master and the unchanged new boot environment.

```
# lucreate -c master_BE
-m /:/dev/dsk/c0t1d0s0:ufs -m /usr:/dev/dsk/c0t1d0s3:ufs \
-n copy_BE
# lustatus
# pkgadd SUNWman
# lumount copy_BE /a
# flarcreate -n test.diff -c -A /a /net/server/export/test.diff
# luumount copy_BE
```

Install the differential archive on clone systems. For procedures about installing an archive, see “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 317.

Figure 21–1 shows the creation of the new boot environment by using the `lucreate` command.



```
# lucreate
-m /:/dev/dsk/c0t1d0s0:ufs \
-n second_disk
```

FIGURE 21–1 Creating a New Boot Environment by Using Solaris Live Upgrade

CHAPTER **22**

Installing and Administering Solaris Flash Archives (Tasks)

This chapter provides references to procedures for installing Solaris Flash archives on clone systems and procedures for administering an archive.

- "References to Procedures for Installing Solaris Flash Archives" on page 237
- "Administering Solaris Flash Archives" on page 238

References to Procedures for Installing Solaris Flash Archives

You can use any of the Solaris installation methods to install Solaris Flash archives for an initial installation. You must use custom JumpStart to install a Solaris Flash differential archive.

- For an initial installation to install a Solaris Flash archive, see the following references.
  - Solaris Live Upgrade – See "Installing Solaris Flash Archives on a Boot Environment" on page 456.
  - The Solaris Web Start program – See "SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program" on page 176 or "x86: Performing an Installation or Upgrade With the Solaris Web Start Program" on page 181.
  - The Solaris `suninstall` program – See "SPARC: Performing an Installation or Upgrade With the Solaris `suninstall` Program" on page 192 or "x86: Performing an Installation or Upgrade With the Solaris `suninstall` Program" on page 195.
  - Custom JumpStart installation program – See "Creating a Profile" on page 277 and "To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation" on page 317.
  - WAN boot installation method – See Chapter 41.

- For an update with a Solaris Flash differential archive, see the following references.
  - Custom JumpStart installation program – See “Creating a Profile” on page 277 and “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 317.
  - Solaris Live Upgrade – See “To Install a Solaris Flash Archive With a Profile (Command-Line Interface)” on page 459.

Administering Solaris Flash Archives

The `flar` command enables you to administer archives. You can split an archive into sections. Those sections can be modified, added to, or deleted, and then merged to create an archive. You can also obtain information about the archive.

Caution – Do not modify the Archive Files section or you compromise the integrity of the archive.

Splitting a Solaris Flash Archive

You can split an archive into sections, which enables you to modify some sections, add new sections, or delete sections. After you have modified the sections, you need to merge the sections to create an new archive. For example, you might want to add a User-Defined section or modify the Archive Identification section. Do not modify the Archive Files section or you compromise the integrity of the archive.

The `flar split` command splits a Solaris Flash archive into sections. The `flar` command copies each section into a separate file in the current directory or specified directory. The files are named after the sections, for example, the archive cookie is saved in a file that is named `cookie`. You can specify that the `flar split` command save only one section. The syntax of the command is as follows:

`flar split`[`-d` *dir*] [`-u` *section*] [`-f` archive] [`-S` *section*] [`-t` [`-p` *posn*] [`-b` *blocksize*]] *filename*

`-d` *dir*	Retrieves the sections to copy from *dir*, rather than from the current directory

`-u` *section*	■ If you use this option, `flar` copies the Cookie, Identification, Archive, and *section* sections. You can specify a single section name or a space-separated list of section names. ■ If you do *not* use this option, `flar` copies the Cookie, Identification, and Archive sections only.
`-f` archive	Extracts the Archive section into a directory that is named `archive`, rather than placing it in a file with the name `archive`
`-S` *section*	Only copies the section that is named *section* from the archive. This section is user-defined.

EXAMPLE 22–1 Splitting an Archive

In the following example, `archive1.flar` is split into three files:

- `cookie` – The first line of the archive, which identifies the version of the archive format. Do not change this identifier.
- `identification` – A copy of the Archive Identification section with all keyword-value pairs.
- `archive` – The cpio archive itself. This file can be compressed.

```
# flar split archive1.flar
```

After the archive is split, you can modify the Archive Identification section or add a User-Defined section. The sections can then be merged to recreate the archive.

Merging a Solaris Flash Archive

After you have split an archive into sections, you can combine the sections to create a new archive.

The `flar combine` command creates a Solaris Flash archive from individual sections. Each section is assumed to be in a separate file, the names of which are the section names. At a minimum, these three files must be present:

- Archive Cookie (`cookie`)
- Archive Identification (`identification`)
- Archive Files (`archive`)

When combining sections, remember the following points:

- If `archive` is a directory, `flar` uses `cpio` to archive the directory before including it in the combined archive.

- If the Archive Identification section specifies to compress the archive, `flar` compresses the contents of the newly combined archive.
- No validation is performed on any of the sections. In particular, no fields in the Archive Identification section are validated or updated.

`flar combine` [`-d` *dir*] [`-u` *section*] [`-t` [`-p` *posn*] [`-b` *blocksize*]] *filename*

`-d` *dir* Retrieves the sections to combine from *dir*, rather than from the current directory.

`-u` *section*

- If you use this option, `flar` copies the Cookie, Identification, Archive, and *section* sections. You can specify a single section name or a space-separated list of section names.
- If you do *not* use this option, `flar` copies the Cookie, Identification, and Archive sections only.

EXAMPLE 22–2 Merging a Solaris Flash Archive

In this example, an Archive Cookie section, an Archive Identification section, and an Archive Files section are combined to become a complete archive. The archive is named `newarchive.flar`.

```
# flar combine newarchive.flar
```

EXAMPLE 22–3 Merging a Solaris Flash Archive and Adding a User-Defined Section

In this example, an Archive Cookie section, an Archive Identification section, an Archive Files section, and a User-Defined section are combined to become a complete archive. The archive is named `newarchive.flar`. The User-Defined section content is in the file that is named `user-defined`, which is in the current directory.

```
# flar combine -u user_defined newarchive.flar
```

Extracting Information From an Archive

Use the `flar info` command to get information about archives you have already created. The syntax of the command is as follows:

`flar info` [`-l`] [`-k` *keyword*] [`-t` [`-p` *posn*] [`-b` *blocksize*]] *filename*

`-k` *keyword* Returns only the value of the keyword *keyword*.

`-l` Lists all the files in the archive section.

EXAMPLE 22–4 Listing Files in an Archive Section

In this example, the file structure of the archive named, `archive3.flar`, is checked.

EXAMPLE 22–4 Listing Files in an Archive Section *(Continued)*

```
# flar info -l archive3.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/eee
```

CHAPTER **23**

Solaris Flash (Reference)

This chapter provides a description of Solaris Flash sections, keywords, and keyword values. Also, the chapter describes the `flar create` command options.

- "Solaris Flash Archive Section Descriptions" on page 243
- "Solaris Flash Keywords" on page 245
- "Solaris Flash `flar create` Command" on page 250

Solaris Flash Archive Section Descriptions

Each Solaris Flash archive is grouped into sections. Some sections are generated by the Solaris Flash software and need no input from you. Some sections require input or optionally allow you to add information. The following table describes each section.

TABLE 23–1 Flash Archive Sections

Section Name	Description	Required by Archive?	Requires Input From User?
Archive cookie	The first section contains a cookie that identifies the file as a Solaris Flash archive. The deployment code uses the cookie for identification and validation purposes. The cookie must be present for an archive to be valid.	Yes	No

TABLE 23–1 Flash Archive Sections *(Continued)*

Section Name	Description	Required by Archive?	Requires Input From User?
Archive identification	The second section contains keywords with values that provide identification information about the archive. The software generates some information such as the following: ■ The archive ID number ■ The method of archival such as cpio ■ The creation date by default You are required to specify a name for the Solaris Flash archive. Other information that you can specify about the archive includes: ■ The author of the archive ■ The date that the archive was created ■ The name of the master system that you used to create the archive For a list of keywords that describe the archive, see "Identification Section Keywords" on page 246.	Yes	Content is generated by both user and the software
Manifest	A section of a Solaris Flash archive that is used to validate a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. The installation fails if the files do not match the expected file set. This section is informational only. The section lists the files in an internal format and cannot be used for scripting. You can exclude this section by creating the differential archive with the `flar create -M` option. Because no validation of the archive occurs, excluding this section is not recommended.	No	No
■ Predeployment ■ Postdeployment ■ Reboot	This section contains internal information that the flash software uses before and after installing an operating environment image. Any customization scripts that you have provided are stored in this section.	Yes	No
Summary	This section contains messages about the archive creation and records the activities of predeployment scripts.	Yes	Content is generated by both user and the software
User-defined	This section follows the identification section. The archive can contain zero or more user-defined sections. These sections are not processed by the archive extraction code. These sections are retrieved separately and can be used for content descriptions.	No	Yes

TABLE 23–1 Flash Archive Sections *(Continued)*

Section Name	Description	Required by Archive?	Requires Input From User?
Archive files	The archive files section contains the files that have been gathered from the master system in binary data. This section begins with `section_begin=archive`, but it does not have an ending section boundary.	Yes	No

Solaris Flash Keywords

Solaris Flash keywords are like custom JumpStart keywords. They define elements of the installation. Each keyword is a command that controls one aspect of how the Solaris Flash software installs the software on a clone system.

Use the following guidelines to format keywords and values:

- Keywords and values are separated by a single equal sign with only one pair per line
- Keywords are case insensitive
- Individual lines can be any length

General Keywords

Each Solaris Flash archive section is defined by the `section_begin` and `section_end` keywords. For example, the archive files section includes a `section_begin` keyword, though with a different value. User-defined archive sections are delimited by `section_begin` and `section_end` keywords, with values appropriate to each section. The values for the `section_begin` and `section_end` keywords are described in the following table.

TABLE 23–2 Values for `section_begin` and `section_end` Keywords

Archive Section	Value for `section_begin` and `section_end` keywords
Archive cookie	`cookie` — This section is not delimited by the `section_begin` and `section_end` keywords.
Archive identification	`identification`
User-defined sections	*section_name* An example of a `section_name` keyword is X-user_section_1.

TABLE 23–2 Values for `section_begin` and `section_end` Keywords *(Continued)*

Archive Section	Value for `section_begin` and `section_end` keywords
Archive files	`archive`

Identification Section Keywords

The following tables describe the keywords for use in the Archive Identification section and the values you can define for them.

Every section uses the keywords in Table 23–3 to delimit each section.

TABLE 23–3 Identification Section Keywords: General Keywords

Keywords	Value Definitions	Value	Required
`section_begin` `section_end`	These keywords are used to delimit sections in the archive and are not limited exclusively to the identification section. For a description of these keywords, see "General Keywords" on page 245.	Text	Yes

The following keywords, used in the archive identification section, describe the contents of the archive files section.

TABLE 23–4 Identification Section Keywords: Contents of Archive Files Section

Keywords	Value Definitions	Value	Required
`archive_id` (optional)	This keyword uniquely describes the contents of the archive. This value is used by the installation software only to validate the contents of the archive during archive installation. If the keyword is not present, no integrity check is performed. For example, the `archive_id` keyword might be `FlAsH-ARcHive-2.0`.	Text	No
`files_archived_method`	This keyword describes the archive method that is used in the files section. ■ If the keyword is present, it has the value of `cpio`. ■ If this keyword is not present, the files section is assumed to be in CPIO format with ASCII headers. This format is the `cpio -c` option. If the `files_compressed_method` is present, the compression method is applied to the archive file that is created by the archive method.	Text	No

TABLE 23–4 Identification Section Keywords: Contents of Archive Files Section *(Continued)*

Keywords	Value Definitions	Value	Required
`files_archived_size`	This keyword value is the size of the archived files section in bytes.	Numeric	No
`files_compress_method`	This keyword describes the compression algorithm that is used on the files section. ■ If the keyword is present, it can have one of the following values. ■ `none` – The archive file section is not compressed ■ `compress` – The file section is compressed by using the `compress` command ■ If this keyword is not present, the archive files section is assumed to be uncompressed. The compression method indicated by this keyword is applied to the archive file created by the archive method indicated by the `files_archived_method` keyword.	Text	No
`files_unarchived_size`	This keyword defines the cumulative size in bytes of the extracted archive. The value is used for file-system size verification.	Numeric	No

The following keywords provide descriptive information about the entire archive. These keywords are generally used to assist you in archive selection and to aid in archive management. These keywords are all optional and are used to help you to distinguish between individual archives. You use options for the `flar create` command to include these keywords. For an example, see Example 21–12.

TABLE 23–5 Identification Section Keywords: User Describes the Archive

Keywords	Value Definitions	Value	Required
`creation_date`	This keyword value is a textual timestamp that represents the time that you created the archive. ■ You can use `flar create` command with the `-i` option to create the date. ■ If you do not specify a creation date with the `flar create` command, the default date is set in Greenwich mean time (GMT). ■ The value must be in ISO-8601 complete basic calendar format without the time designator (ISO-8601,§5.4.1(a)). The format is *CCYYMMDDhhmmss*. For example, 20000131221409 represents January 31st, 2000 10:14:09 p.m.	Text	No

TABLE 23–5 Identification Section Keywords: User Describes the Archive *(Continued)*

Keywords	Value Definitions	Value	Required
`creation_master`	This keyword value is the name of the master system you used to create the archive. You can use the `flar create -m` option to create this value. If you do not specify a value, the value is taken from the `uname -n` command.	Text	No
`content_name`	This keyword identifies the archive. The value is generated from the `flar create -n` option. Follow these guidelines when you create this value: ■ The descriptive name can be no longer than 256 characters. ■ The description should contain the function and purpose of the archive.	Text	Yes
`content_type`	This keyword value specifies a category for the archive. You use the `flar create -T` option to generate the value.	Text	No
`content_description`	The keyword value describes the contents of the archive. The value of this keyword has no length limit. You use the `flar create -E` option to create this value.	Text	No
`content_author`	This keyword value identifies the creator of the archive. You use the `flar create-a` option to create this value. Suggested values include the full name of the creator and the creator's email address.	Text	No
`content_architectures`	This keyword value is a comma-separated list of the kernel architectures that the archive supports. ■ If the keyword is present, the installation software validates the kernel architecture of the clone system against the list of architectures that the archive supports. The installation fails if the archive does not support the kernel architecture of the clone system. ■ If the keyword is not present, the installation software does not validate the architecture of the clone system.	Text list	No

The following keywords also describe the entire archive. By default, the values are filled in by `uname` when the flash archive is created. If you create a flash archive in which the root directory is not `/`, the archive software inserts the string UNKNOWN for the keywords. The exceptions are the `creation_node`, `creation_release`, and `creation_os_name` keywords.

- For `creation_node`, the software uses the contents of the `nodename` file.

- For `creation_release` and `creation_os_name`, the software attempts to use the contents of `root` directory `/var/sadm/system/admin/INST_RELEASE`. If the software is unsuccessful in reading this file, it assigns the value UNKNOWN.

Regardless of their sources, you cannot override the values of these keywords.

TABLE 23–6 Identification Section Keywords: Software Describes the Archive

Keyword	Value
`creation_node`	The return from `uname -n`
`creation_hardware_class`	The return from `uname -m`
`creation_platform`	The return from `uname -i`
`creation_processor`	The return from `uname -p`
`creation_release`	The return from`uname -r`
`creation_os_name`	The return from `uname -s`
`creation_os_version`	The return from `uname -v`

User-Defined Section Keywords

In addition to the keywords that are defined by the Solaris Flash archive, you can define other keywords. The Solaris Flash archive ignores user-defined keywords, but you can provide scripts or programs that process the identification section and use user-defined keywords. Use the following format when creating user-defined keywords:

- Begin the keyword name with an `X`.
- Create the keyword with any characters other than linefeeds, equal signs, and null characters.
- Suggested naming conventions for user-defined keywords include the underscore-delimited descriptive method used for the predefined keywords. Another convention is a federated convention similar to the naming of Java packages.

For example, `X-department` is a valid name for a user-defined keyword.

For an example of using options to include user-defined keywords in the identification section, see Example 21–12.

Solaris Flash `flar create` Command

Use the Solaris Flash `flar create` command to create a Solaris Flash archive.

`flar create`

Use the `flar create` command to create a Solaris Flash archive from a master system. You can use this command when the master system is running in multiuser mode or single-user mode. You can also use `flar create` when the master system is booted from the Solaris DVD or the Solaris Software 1 of 2 CD or from an image of the Solaris Software and Solaris Languages CDs. The master system should be in as stable a state as possible when you create a Solaris Flash archive. The syntax of the command is as follows:

```
flar create -n archive_name [-R root] [-A unchanged_master_image_dir] [-S]
[-M] [-H] [-I] [-c] [-x exclude_dir/filename] [-y include_dir/filename] [-z
list_filename] [-X list_filename] [-t [-p posn] [-b blocksize] [-i date] [-m
master] [-u section ... [-d dir]] [-f [list_filename| -] [-F]] [-U key=val
...] [-a author] [-e descr|-E descr_file] [-T type] path/filename
```

In this command line, *path* is the directory in which you want the archive file to be saved. *filename* is the name of the archive file. If you do not specify a path, `flar create` saves the archive file in the current directory.

TABLE 23–7 Command-Line Options for `flar create`

Option	Description
Required Options	
`-n` *archive_name*	The value of this flag is the name of the archive. The *archive_name* you specify is the value of the `content_name` keyword.
Option for Compression	
`-c`	Compresses the archive by using `compress(1)`.
Options for Directories and Sizes	
`-R` *root*	Creates the archive from the file system tree that is rooted at *root*. If you do not specify this option, `flar create` creates an archive from a file system that is rooted at `/`.
`-S`	Omits sizing information in the archive.

TABLE 23–7 Command-Line Options for `flar create` *(Continued)*

Option	Description
`-H`	Does not generate the hash identifier.
Options for Creating a Differential Archive	
`-A` *unchanged_master_image_dir*	Create a differential archive by comparing a new system image with the image that is specified by the *unchanged_master_image_dir* argument. By default, the new system image is root (`/`). You can change the default with the `-R` option. *unchanged_master_image_dir* is a directory where the unchanged master system image is stored or mounted through UFS, NFS, or `lumount`. You can modify the effects of file selection for a differential archive by using the options for contents selection described in the next section of the table.
`-M`	Excludes the manifest file. When you use this option, no validation occurs on the differential archive. When creating a differential archive, `flar create` creates a long list of the files in the system that are unchanged, are changed, and are to be deleted from the archive. This list is stored in the manifest section of the archive. When the differential archive is deployed, the software uses this list to perform a file-by-file check, ensuring the integrity of the clone system. Use of this option avoids such a check and saves the space that is used by the manifest section in a differential archive. However, you must consider the savings in time and disk space against the loss of an integrity check upon installation. Because no validation occurs, avoid using this option.
Options for Contents Selection	
Caution – Use the `flar create` file-exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.	
`-y` *include_dir/filename*	Adds to the archive those files and directories that are specified on the command line. This option is used when you've excluded a directory, but want to restore individual subdirectories or files. *include_dir/filename* is the name of the subdirectory or file to be included.

TABLE 23–7 Command-Line Options for `flar create` *(Continued)*

Option	Description
`-f` *list_filename*	Adds files and directories from a list to the archive. *list_filename* is the full path to a file that contains a list. The contents of the file are added to the file list unless `-F` is specified. ■ The *list_filename* file must contain one file per line. ■ If you specify a file system with `-R` *root*, the path to each file must be relative to the alternate `root` directory or an absolute path. ■ If *filename* is "-", `flar create` reads standard input as the list of files. When you use the value "-", the archive size is not calculated.
`-F`	Uses only the files in `-f` *list_filename* to create the archive. This option makes the `-f` *list_filename* the absolute list, rather than a list that is appended to the normal file list.
`-x` *exclude_dir/filename*	Excludes files and directories from the archive. These files and directories are specified at the command line. You can use multiple instances of this option to exclude more than one file or directory. *exclude_dir/filename* is the name of the directory or file to be excluded.
`-X` *list_filename*	Excludes a list of files or directories from the archive. *list_filename* is the full path to a file that contains the list. ■ The *list_filename* file must contain one file per line. ■ If you specify a file system with `-R` *root*, the path to each file must be relative to the alternate `root` directory or an absolute path. ■ If *list_filename* is "-", `flar create` reads standard input as the list of files. When you use the value "-", the archive size is not calculated.
`-z` *list_filename*	Excludes or includes a list of files or directories from the archive. Each file or directory in the list is noted with a plus "+" or minus "-". A plus indicates an included file or directory and the minus indicates an excluded file or directory. *list_filename* is the full path to a file that contains the list. ■ The *list_filename* file must contain one file per line. ■ If you specify a file system with `-R` *root*, the path to each file must be relative to the alternate `root` directory or an absolute path.

TABLE 23–7 Command-Line Options for `flar create` *(Continued)*

Option	Description
`-I`	Override the integrity check. To prevent you from excluding important system files from an archive, `flar create` runs an integrity check. This check examines all files that are registered in a system package database and stops archive creation if any of them are excluded. Use of this option overrides this integrity check. Therefore, avoid the use of the `-I` option.
Options Used With User-Defined Sections	
`-u` *section*	Includes *section* as a user-defined section. To include more than one user-defined section, *section* must be a space-separated list of section names.
`-d` *dir*	Retrieves the section file that is specified with `-u` from *dir*.
Options Used With Tape Archives	
`-t`	Creates an archive on a tape device. The *filename* argument is the name of the tape device.
`-p` *posn*	Use only with the `-t` option. Specifies the position on the tape device for `flar create` to store the archive. If you do not use this option, `flar create` places the archive at the current position of the tape.
`-b` *blocksize*	Specifies the block size `flar create` uses when creating the archive. If you do not specify a block size, `flar create` uses the default block size of 64k.
Options for Archive Identification These keywords and values appear in the identification section of the archive.	
`-U` *key=val*	Includes user-defined keywords and values in the Archive Identification section.
`-i` *date*	Uses *date* as the value for the `creation_date` keyword. If you do not specify a date, `flar create` uses the current system time and date.
`-m` *master*	Uses *master* as the name of the master system on which you created the archive. *master* is the value for the `creation_master` keyword. If you do not specify *master*, `flar create` uses the system name that is reported by `uname -n`.
`-e` *descr*	Uses *descr* for the value of the `content_description` keyword. You cannot use this option when you use the `-E` option.

TABLE 23–7 Command-Line Options for `flar create` *(Continued)*

Option	Description
`-E` *descr_file*	Retrieves the value for the `content_description` keyword from the *descr_file* file. You cannot use this option when you use the `-e` option.
`-a` *author*	Uses *author* as the author name in the Archive Identification section. *author* is the value for the `content_author` keyword. If you do not specify an author, `flar create` does not include the `content_author` keyword in the Archive Identification section.
`-T` *type*	Uses *type* as the value for the `content_type` keyword. *type* is user defined. If you do not specify a type, `flar create` does not include the `content_type` keyword.

CHAPTER **24**

Custom JumpStart Installation (Topics)

This section provides instructions for creating, preparing, and performing custom JumpStart installations.

Chapter 25	Provides an introduction and overview of the custom JumpStart installation method.
Chapter 26	Provides instructions about how to prepare the systems at your site from which and on which you are installing the Solaris 9 software with the custom JumpStart installation method.
Chapter 27	Describes the optional features that you can use to create additional tools for a custom JumpStart installation.
Chapter 28	Provides information and procedures for creating your own custom rule and custom probe keywords.
Chapter 29	Describes how to perform a custom JumpStart installation on a SPARC based or an x86 based system. You need to follow these procedures on the system on which you intend to install the Solaris 9 software.
Chapter 30	Provides an example of setting up and installing Solaris software on both SPARC based and x86 based systems by using the custom JumpStart installation method.
Chapter 31	Contains lists of keywords and values to be used in the `rules` file, profiles, `begin` scripts, and `finish` scripts.

CHAPTER 25

Custom JumpStart (Overview)

This chapter provides an introduction and overview to the custom JumpStart installation process.

Custom JumpStart Introduction

The custom JumpStart installation method is a command–line interface that enables you to automatically install or upgrade several systems, based on profiles that you create. The profiles define specific software installation requirements. You can also incorporate shell scripts to include preinstallation and postinstallation tasks. You choose which profile and scripts to use for installation or upgrade. The custom JumpStart installation method installs or upgrades the system, based on the profile and scripts that you select. Also, you can use a `sysidcfg` file to specify configuration information so that the custom JumpStart installation is completely hands-off.

The custom JumpStart process can be described by using an example scenario. In this example scenario, the systems need to be set up with the following parameters:

- Install Solaris on 100 new systems.
- Seventy of the systems are SPARC systems that are owned by the engineering group and need to be installed as standalone systems with the Solaris operating environment software group for developers.
- The remaining 30 systems are x86 based, owned by the marketing group and need to be installed as standalone systems with the Solaris operating environment software group for end users.

First, the system administrator must create a rules file and a profile for each group of systems. The rules file is a text file that contains a rule for each group of systems or single systems on which you want to install the Solaris software. Each rule distinguishes a group of systems that are based on one or more system attributes. Each rule also links each group to a profile.

A profile is a text file that defines how the Solaris software is to be installed on each system in the group. Both the rules file and profile must be located in a JumpStart directory.

For the example scenario, the system administrator creates a rules file that contains two different rules, one for the engineering group and another for the marketing group. For each rule, the system's network number is used to distinguish the engineering group from the marketing group.

Each rule also contains a link to an appropriate profile. For example, in the rule for the engineering group, a link is added to the profile, eng_profile, which was created for the engineering group. In the rule for the marketing group, a link is added to the profile, market_profile, which was created for the marketing group.

You can save the rules file and the profiles on a diskette or on a server.

- A profile diskette is required when you want to perform custom JumpStart installations on non-networked, standalone systems.
- A profile server is used when you want to perform custom JumpStart installations on networked systems that have access to a server.

After creating the rules file and profiles, validate the files with the check script. If the check script runs successfully, the rules.ok file is created. The rules.ok is a generated version of the rules file that the JumpStart program uses to install the Solaris software.

How the JumpStart Program Installs Solaris Software

After you validate the rules file and the profiles, you can begin a custom JumpStart installation. The JumpStart program reads the rules.ok file. Then, the JumpStart program searches for the first rule with defined system attributes that match the system on which the JumpStart program is attempting to install the Solaris software. If a match occurs, the JumpStart program uses the profile that is specified in the rule to install the Solaris software on the system.

Figure 25–1 illustrates how a custom JumpStart installation works on a standalone, non-networked system. The system administrator initiates the custom JumpStart installation on Pete's system. The JumpStart program accesses the rules files on the diskette in the system's diskette drive. The JumpStart program matches `rule 2` to the system. `rule 2` specifies that the JumpStart program use `Pete's profile` to install the Solaris software. The JumpStart program reads `Pete's profile` and installs the Solaris software, based on the instructions that the system administrator specified in `Pete's profile`.

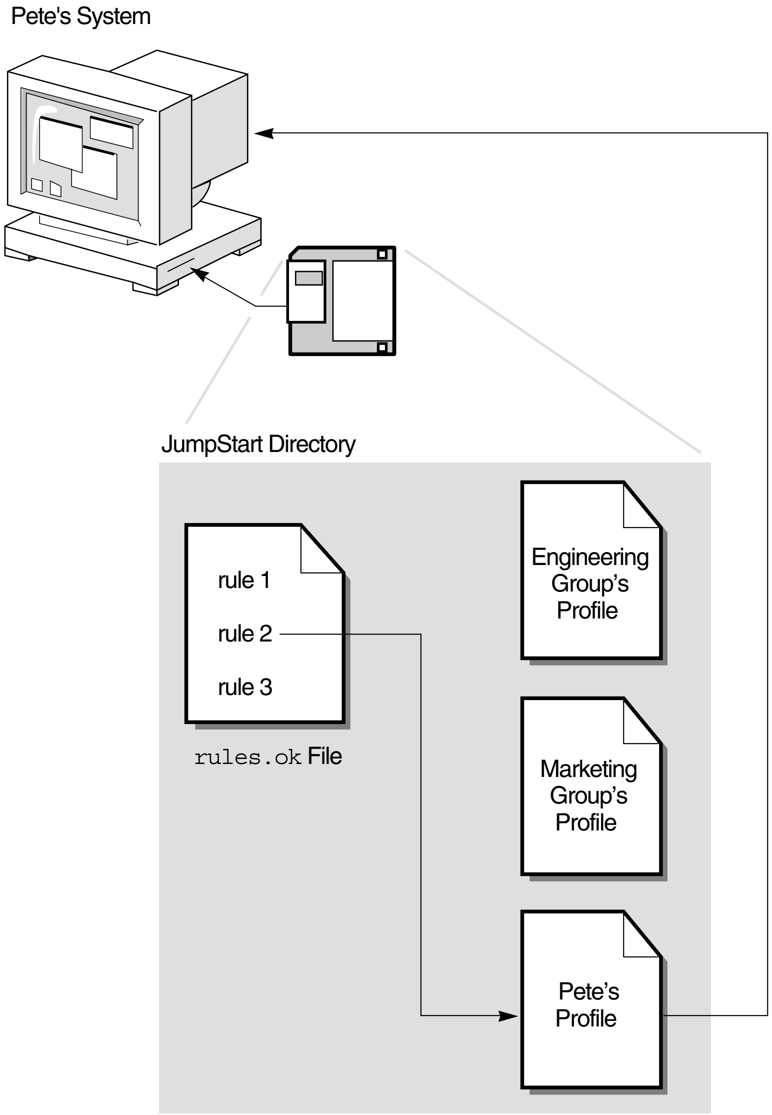


FIGURE 25–1 How a Custom JumpStart Installation Works: Non-Networked Example

Figure 25–2 illustrates how a custom JumpStart installation works with more than one system on a network. The system administrator set up different profiles and saved the profiles on a single server. The system administrator initiates the custom JumpStart installation on one of the engineering systems. The JumpStart program accesses the rules files in the `JumpStart/` directory on the server. The JumpStart program matches the engineering system to `rule 1`. `rule 1` specifies that the JumpStart program use `Engineering Group's Profile` to install the Solaris software. The JumpStart program reads `Engineering Group's Profile` and installs the Solaris software, based on the instructions that the system administrator specified in `Engineering Group's Profile`.

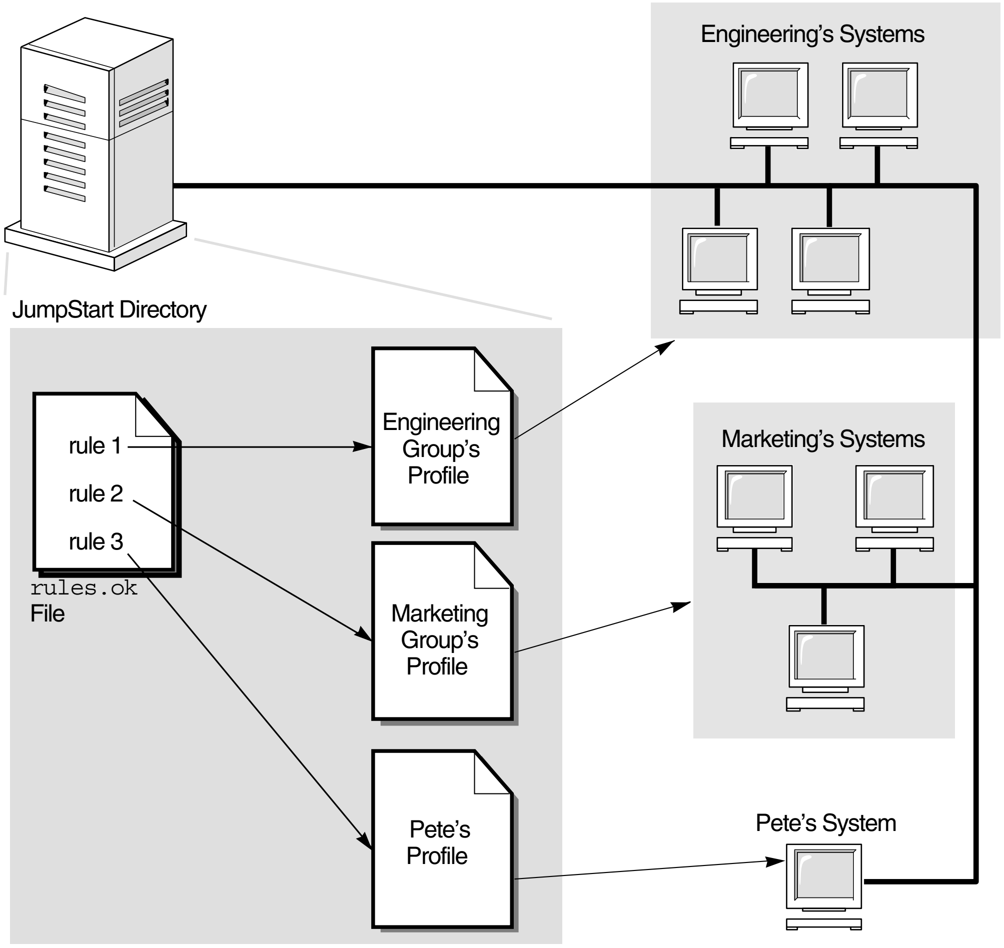


FIGURE 25–2 How a Custom JumpStart Installation Works: Networked Example

Figure 25–3 describes the order in which the JumpStart program searches for custom JumpStart files.

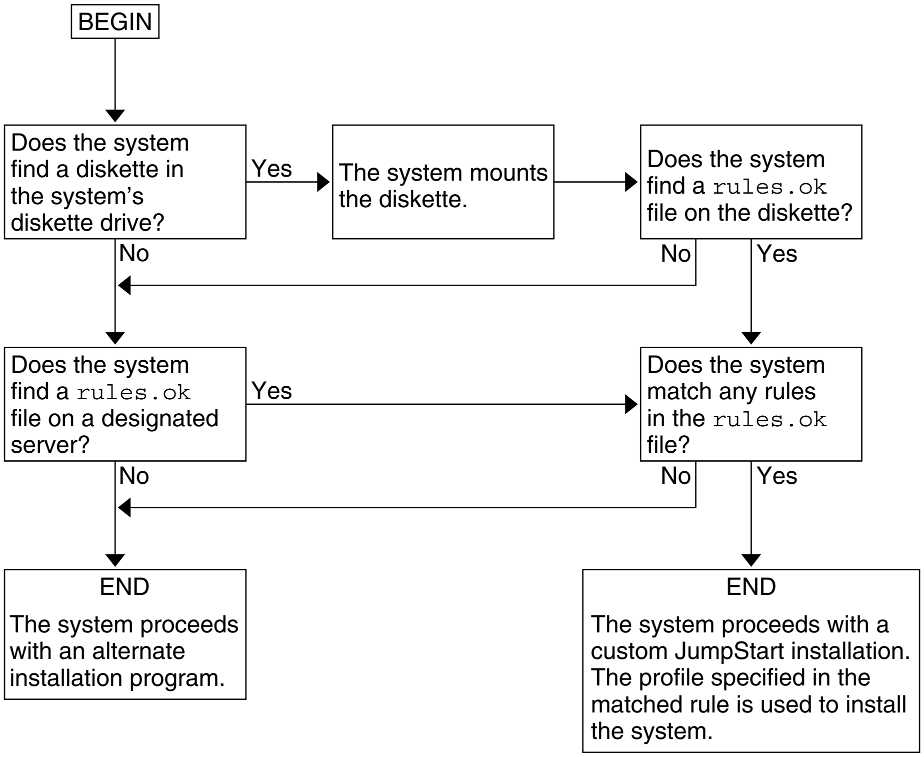

BEGIN
Does the system find a diskette in the system's diskette drive?
Yes
No
The system mounts the diskette.
Does the system find a rules.ok file on the diskette?
No
Yes
Does the system find a rules.ok file on a designated server?
Yes
No
Does the system match any rules in the rules.ok file?
No
Yes
END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

FIGURE 25–3 What Happens During a Custom JumpStart Installation

CHAPTER **26**

Preparing Custom JumpStart Installations (Tasks)

This chapter provides step-by-step instructions about how to prepare the systems at your site from which and on which you intend to install the Solaris 9 software by using the custom JumpStart installation method.

- "Task Map: Preparing Custom JumpStart Installations" on page 263
- "Creating a Profile Server for Networked Systems" on page 265
- "Creating a Profile Diskette for Standalone Systems" on page 269
- "Creating the `rules` File" on page 274
- "Creating a Profile" on page 277
- "Testing a Profile" on page 286
- "Validating the `rules` File" on page 290

Task Map: Preparing Custom JumpStart Installations

TABLE 26–1 Task Map: Preparing Custom JumpStart Installations

Task	Description	For Instructions
Decide how to upgrade the system if a previous version of the Solaris software is installed on the system	If a previous release of Solaris is installed on the system, you need to determine how to upgrade the system. Ensure that you know what to do before and after you upgrade a system. Planning helps you to create your profiles, begin scripts, and finish scripts.	Chapter 8

TABLE 26–1 Task Map: Preparing Custom JumpStart Installations *(Continued)*

Task	Description	For Instructions
Create a JumpStart directory	**On a server** If you want to perform custom JumpStart installations on systems that are connected to a network, you must create a profile server. The profile server contains a JumpStart directory for the custom JumpStart files.	"Creating a Profile Server for Networked Systems" on page 265
	On a diskette If you want to perform custom JumpStart installations on systems that are not connected to a network, you must create a profile diskette. A profile diskette contains the custom JumpStart files.	"Creating a Profile Diskette for Standalone Systems" on page 269
Add rules to the `rules` file	After you decide how you want each group of systems or single systems to be installed, create a rule for each group that you want to install. Each rule distinguishes a group, based on one or more system attributes. The rule links each group to a profile.	"Creating the `rules` File" on page 274
Create a profile for every rule	A profile is a text file that defines how to install the Solaris software, for example, which software group to install on a system. Every rule specifies a profile to define how a system is to be installed with the Solaris software when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule.	"Creating a Profile" on page 277
(Optional) Test the profiles	After you create a profile, use the `pfinstall`(1M) command to test the profile before you use the profile to install or upgrade a system.	"Testing a Profile" on page 286

TABLE 26–1 Task Map: Preparing Custom JumpStart Installations *(Continued)*

Task	Description	For Instructions
Validate the `rules` file	The `rules.ok` file is a generated version of the `rules` file that the JumpStart program uses to match the system to be installed with a profile. You must use the `check` script to validate the `rules` file.	"Validating the `rules` File" on page 290

Creating a Profile Server for Networked Systems

When setting up custom JumpStart installations for systems on the network, you need to create a directory on a server that is called a JumpStart directory. The JumpStart directory contains all of the essential custom JumpStart files, for example, the `rules` file, `rules.ok` file, and profiles. You must save the JumpStart directory in the root (`/`) directory of the profile server.

The server that contains a JumpStart directory is called a profile server. A profile server can be the same system as an install server or a boot server, or the server can be a completely different server. A profile server can provide custom JumpStart files for different platforms. For example, an x86 server can provide custom JumpStart files for both SPARC systems and x86 systems.

Note – After you create a profile server, you must allow systems to access the server. For detailed instructions, see "To Allow All Systems Access to the Profile Server" on page 267.

▼ To Create a JumpStart Directory on a Server

Note – This procedure assumes that the system is running *Volume Manager*. If you are not using Volume Manager to manage discs, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without Volume Manager.

1. **Log in as superuser on the server on which you want to create the JumpStart directory.**

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

   ```
   # mkdir -m 755 jumpstart_dir_path
   ```

   In the command, *jumpstart_dir_path* is the absolute path of the JumpStart directory.

   For example, the following command creates a directory that is called `jumpstart` in the root (/) directory and sets the permissions to 755:

   ```
   # mkdir -m 755 /jumpstart
   ```

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

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

   For example, the following entry shares the `/jumpstart` directory:

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

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

5. **Determine if you want to copy examples of custom JumpStart files to your JumpStart directory.**

   - If no, go to Step 8.
   - If yes, use the following decision table to determine what to do next.

Example Locations	Instructions
The Solaris DVD or the Solaris Software 1 of 2 CD for your platform	Insert the Solaris DVD or the Solaris Software 1 of 2 CD into the server's CD-ROM drive. Volume Manager automatically mounts the CD.
An image of the Solaris DVD or the Solaris Software 1 of 2 CD for your platform on a local disk	Change directory to the location of the Solaris DVD or the Solaris Software 1 of 2 image. For example, type the following command: `cd /export/install`

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

   ```
   # cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
   ```

   media_path The path to the CD, DVD, or image on the local disk

   jumpstart_dir_path The path on the profile server where you are placing the example custom JumpStart files

   For example, the following command copies the `jumpstart_sample` directory into the `/jumpstart` directory on the profile server:

   - For SPARC systems:

     ```
     cp -r /cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample/* /jumpstart
     ```

- For x86 systems:

  ```
  cp -r /cdrom/cdrom0/s2/Solaris_9/Misc/jumpstart_sample/* /jumpstart
  ```

7. **Update the example JumpStart files so that the files work in your environment.**

8. **Ensure that `root` owns the JumpStart directory and that the permissions are set to 755.**

9. **Allow systems on the network to access the profile server.**

   For detailed instructions, see “To Allow All Systems Access to the Profile Server” on page 267.

▼ To Allow All Systems Access to the Profile Server

When you create a profile server, you must ensure that systems can access the JumpStart directory on the profile server during a custom JumpStart installation. Use one of the following ways to ensure access:

- `add_install_client` command – Each time that you add a system for network installation, use the `-c` option with the `add_install_client` command. For detailed instructions, refer to “Adding Systems to Be Installed From the Network With a CD Image” on page 167.
- `boot` command – Specify the location of the JumpStart directory on the profile server when you boot the system. You must compress the custom JumpStart configuration files into one file. Then, save the compressed configuration file on an NFS server, an HTTP server, or on media that the system can access locally. For detailed instructions, refer to “Creating a Compressed Configuration File” on page 301.

  When you boot the system to initiate the custom JumpStart installation, specify the location of the compressed file. For detailed instructions for SPARC systems, refer to Step 5 in “SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 318. For x86 systems, refer to Step 8 in “x86: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 322.
- `/etc/bootparams` file – Use a wildcard in the `/etc/bootparams` file. Use the following steps to add a wildcard in the `etc/bootparams` file.

Note – The following procedure is not necessary if you save the JumpStart directory on a diskette or if you specify the location of the profile server when you boot the system.

The following procedure is valid only if you store network installation information in the `/etc/bootparams` file. You can also store network installation information in one of the following places:

- Name service database – If you store network installation information in the name service `bootparams` database, you must update the `bootparams` database with the entry that is shown in Step 3.
- DHCP server – If you store network installation information on a DHCP server, use the `boot` command to specify that the custom JumpStart program use the DHCP server. For detailed instructions for SPARC systems, refer to Step 5 in “SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 318. For x86 systems, refer to Step 8 in “x86: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 322.

1. **On the install or boot server, log in as superuser.**
2. **Use a text editor to open `/etc/bootparams`.**
3. **Add this entry.**

   `* install_config=`*server:jumpstart_dir_path*

   *   A wildcard character that specifies that all systems have access

   server   The host name of the profile server where the JumpStart directory is located

   jumpstart_dir_path   The absolute path of the JumpStart directory

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

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

Caution – Use of this procedure might produce the following error message when an install client is booted:

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

“Booting From the Network, Error Messages” on page 635 contains details about this error message.

All systems can now access the profile server.

Creating a Profile Diskette for Standalone Systems

A diskette that contains a JumpStart directory is called a profile diskette. A system that is not connected to the network does not have access to a profile server. As a result, you must create a JumpStart directory on a diskette if a system is not connected to a network. The system on which you create a profile diskette must have a diskette drive.

The JumpStart directory contains all of the essential custom JumpStart files, for example, the `rules` file, `rules.ok` file, and profiles. You must save the JumpStart directory in the root (/) directory of the profile diskette.

▼ SPARC: To Create a Profile Diskette

Note – This procedure assumes that the system is running Volume Manager. If you are not using Volume Manager to manage diskettes, CDs, and DVDs, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without Volume Manager.

---

1. **Log in as superuser on a SPARC system to which a diskette drive is attached.**
2. **Insert a blank diskette or a diskette that can be overwritten in the diskette drive.**
3. **Mount the diskette.**

   ```
   # volcheck
   ```

4. **Determine if the diskette contains a UNIX file system (UFS).**

   Examine the contents of the file `/etc/mnttab` on the system for an entry such as the following:

```
/vol/dev/diskette0/scrap  /floppy/scrap  ufs  suid,rw,largefiles,dev=1740008  927147040
```

   - If the entry exists, go to Step 7.
   - If the entry does not exist, go to the next step.

5. **Format the diskette.**

Caution – Formatting erases all data on the diskette.

```
# fdformat -U
```

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

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

7. **Determine if you want to copy examples of custom JumpStart files to your JumpStart directory.**

- If no, go to Step 10.
- If yes, use the following decision table to determine what to do next.

Example Locations	Instructions
The Solaris *SPARC Platform Edition* DVD or the Solaris Software 1 of 2 *SPARC Platform Edition* CD	Insert the Solaris *SPARC Platform Edition* DVD or the Solaris Software 1 of 2 *SPARC Platform Edition* CD into the server's CD-ROM drive. Volume Manager automatically mounts the CD.
An image of the Solaris *SPARC Platform Edition* DVD or the Solaris Software 1 of 2 *SPARC Platform Edition* CD on a local disk	Change the directory to the location of the Solaris *SPARC Platform Edition* DVD or the Solaris Software 1 of 2 *SPARC Platform Edition* CD image. For example, type the following command: **`cd /export/install`**

8. **Copy the example custom JumpStart files into the JumpStart directory on the profile diskette.**

```
# cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
```

media_path The path to the CD, DVD, or image on the local disk

jumpstart_dir_path The path to the profile diskette where you want to place the example custom JumpStart files

Note – You must place all custom JumpStart installation files in the root (/) directory on the diskette.

For example, the following command copies the contents of `jumpstart_sample` on the Solaris Software 1 of 2 *SPARC Platform Edition* CD to the root (/) directory on a profile diskette that is named `scrap`:

```
cp -r /cdrom/sol_9_sparc/s0/Solaris_9/Misc/jumpstart_sample/* /floppy/scrap
```

9. **Update the example JumpStart files on the profile diskette so that the files work in your environment.**

10. **Ensure that `root` owns the JumpStart directory and that permissions are set to 755.**

11. **Eject the diskette.**

    ```
    # eject floppy
    ```

    You have completed the creation of a profile diskette. You can now update the `rules` file and create profiles on the profile diskette to perform custom JumpStart installations. To continue, go to "Creating the `rules` File" on page 274.

▼ x86: To Create a Profile Diskette

Note – This procedure assumes that the system is running Volume Manager. If you are not using Volume Manager to manage diskettes, CDs, and DVDs, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without Volume Manager.

---

1. **Log in as superuser on an x86 system to which a diskette drive is attached.**

2. **Insert the Solaris Device Configuration Assistant** *x86 Platform Edition* **diskette into the diskette drive (usually drive A). You use this diskette as the profile diskette.**

    ---

    x86 only – You can copy the Device Configuration Assistant software to a diskette from the Solaris *x86 Platform Edition* DVD or Solaris Software 2 of 2 *x86 Platform Edition* CD by using the procedure in Appendix I.

    ---

3. **Mount the diskette.**

    ```
    # volcheck
    ```

4. **Copy the image of the Solaris Device Configuration Assistant to the system's hard disk.**

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

In the command, *boot_image* is the name of the file into which you want to copy the image of the Solaris Device Configuration Assistant. You can specify an absolute path name.

For example, the following command copies the boot diskette to a file that is named `boot_save`:

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

5. **Eject the diskette by clicking Eject Disk in the File Manager window or by typing `eject floppy` on the command line.**

6. **In the Removable Media Manager dialog box, click OK.**

7. **Manually eject the Solaris Device Configuration Assistant** *x86 Platform Edition* **diskette.**

8. **Insert a blank diskette or a diskette that can be overwritten into the diskette drive.**

9. **Mount the diskette.**

```
# volcheck
```

10. **Format the diskette.**

Caution – Formatting erases all data on the diskette.

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

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

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

In the command, *boot_image* is the name of the file where you want to copy the image of the Solaris Device Configuration Assistant. You can specify an absolute path name.

12. **Determine if you want to copy examples of custom JumpStart files to your JumpStart directory.**

- If no, go to Step 15.
- If yes, use the following decision table to determine what to do next.

Example Locations	Instructions
The Solaris *x86 Platform Edition* DVD or the Solaris Software 1 of 2 *x86 Platform Edition* CD	Insert the Solaris *x86 Platform Edition* DVD or the Solaris Software 1 of 2 *x86 Platform Edition* CD into the server's CD-ROM drive. Volume Manager automatically mounts the CD.
An image of the Solaris *x86 Platform Edition* DVD or the Solaris Software 1 of 2 *x86 Platform Edition* CD on a local disk	Change directory to the location of the Solaris *x86 Platform Edition* DVD or the Solaris Software 1 of 2 *x86 Platform Edition* CD image. For example, type the following: `cd /export/install`

13. **Copy the example custom JumpStart files into the JumpStart directory on the profile diskette.**

```
# cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
```

media_path The path to the CD, DVD, or image on the local disk

jumpstart_dir_path The path to the profile diskette where you want to place the example custom JumpStart files

Note – You must place all custom JumpStart installation files in the root (/) directory on the profile diskette.

For example, the following command copies the contents of `jumpstart_sample` on the Solaris Software 1 of 2 *x86 Platform Edition* CD to the root (/) directory on a profile diskette that is named `scrap`:

```
cp -r /cdrom/sol_9_x86/s2/Solaris_9/Misc/jumpstart_sample/* /floppy/scrap
```

14. **Update the example JumpStart files on the profile diskette so that the files work in your environment.**

15. **Ensure that `root` owns the JumpStart directory and that permissions are set to 755.**

16. **Eject the diskette by clicking Eject Disk in the File Manager window or by typing `eject floppy` on the command line.**

17. **In the Removable Media Manager dialog box, click OK.**

18. **Manually eject the diskette.**

You have completed the creation of a profile diskette. Now you can update the `rules` file and create profiles on the profile diskette to perform custom JumpStart installations. To continue, go to "Creating the `rules` File" on page 274.

Creating the rules File

The rules file is a text file that contains a rule for each group of systems on which you want to install the Solaris operating environment. Each rule distinguishes a group of systems that are based on one or more system attributes. Each rule also links each group to a profile. A profile is a text file that defines how the Solaris software is to be installed on each system in the group. For example, the following rule specifies that the JumpStart program use the information in the basic_prof profile to install any system with the sun4u platform group.

```
karch sun4u - basic_prof -
```

The rules file is used to create the rules.ok file, which is required for custom JumpStart installations.

Note – If you set up the JumpStart directory by using the procedures in "Creating a Profile Diskette for Standalone Systems" on page 269 or "Creating a Profile Server for Networked Systems" on page 265, an example rules file is already located in the JumpStart directory. The sample rules file contains documentation and some example rules. If you use the sample rules file, ensure that you comment out the example rules you do not intend to use.

Syntax of the rules File

The rules file must have the following attributes:

- The file must be assigned the name rules.
- The file must contain at least one rule.

The rules file can contain any of the following:

- Commented text

  Any text that is included after the # symbol on a line is treated by JumpStart as commented text. If a line begins with the # symbol, the entire line is treated as a comment.
- One or more blank lines
- One or more multiline rules

  To continue a single rule onto a new line, include a backslash character (\) just before pressing Return.

▼ To Create a `rules` File

1. **Use a text editor to create a text file that is named `rules`. Or, open the sample `rules` file in the JumpStart directory that you created.**

2. **Add a rule in the `rules` file for each group of systems on which you want to install the Solaris software.**

   For a list of `rules` file keywords and values, see "Rule Keywords and Values" on page 335.

   A rule within a `rules` file must adhere to the following syntax:

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

TABLE 26–2 Syntax Elements of a Rule

Element	Description
`!`	A symbol that is used before a keyword to indicate negation.
rule_keyword	A predefined lexical unit or word that describes a general system attribute, such as host name, `hostname`, or memory size, `memsize`. *rule_keyword* is used with the rule value to match a system with the same attribute to a profile. For the list of rule keywords, see "Rule Keywords and Values" on page 335.
rule_value	A value that provides the specific system attribute for the corresponding rule keyword. Rule values are described in "Rule Keywords and Values" on page 335.
`&&`	A symbol you must use to join rule keyword and rule value pairs in the same rule (a logical AND). During a custom JumpStart installation, a system must match every pair in the rule before the rule matches.
begin	The name of an optional Bourne shell script that can be executed before the installation begins. If no begin script exists, you must type a minus sign (-) in this field. All begin scripts must be located in the JumpStart directory. Information about how to create begin scripts is presented in "Creating Begin Scripts" on page 293.
profile	The name of a text file that defines how the Solaris software is to be installed on the system when a system matches the rule. The information in a profile consists of profile keywords and their corresponding profile values. All profiles must be located in the JumpStart directory. **Note –** Optional ways to use the profile field are described in "Using a Site-Specific Installation Program" on page 307 and "Creating Derived Profiles With a Begin Script" on page 294.

TABLE 26–2 Syntax Elements of a Rule *(Continued)*

Element	Description
finish	The name of an optional Bourne shell script that can be executed after the installation is completed. If no finish script exists, you must type a minus sign (-) in this field. All finish scripts must be located in the JumpStart directory. Information about how to create finish scripts is presented in “Creating Finish Scripts” on page 295.

At the minimum, each rule must contain the following:

- A keyword, a value, and a corresponding profile
- A minus sign (-) in the *begin* and *finish* fields if no begin or finish scripts are specified

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

4. **Ensure that `root` owns the `rules` file and that the permissions are set to 644.**

`rules` File Example

The following example shows several example rules in a `rules` file. Each line has a rule keyword and a valid value for that keyword. The JumpStart program scans the `rules` file from top to bottom.

When the JumpStart program matches a rule keyword and value with a known system, the JumpStart program installs the Solaris software that is specified by the profile that is listed in the profile field.

EXAMPLE 26–1 `rule` File

```
 # rule keywords and rule values       begin script       profile       finish script
 # -----------------------------       ------------       --------      -------------
  hostname eng-1                       -                  basic_prof    -
  network 172.31.34.0 && !model \
’SUNW,SPARCstation-20’                 -                  net_prof      -
  model SUNW,SPARCstation-LX           -                  lx_prof       complete
  network 192.168.2.0 && karch i86pc  setup               x86_prof      done
  memsize 64-128 && arch i386          -                  prog_prof     -
  any   -                              -                  generic_prof  -
```

The following list describes some of the keywords and values from this example.

`hostname` The 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 the rule.

`network` The rule matches if the system is on subnet 172.31.34.0 and if the system is *not* a SPARCstation™ 20 (`SUNW,SPARCstation-20`). The

EXAMPLE 26–1 `rule` File *(Continued)*

`net_prof` profile is used to install the Solaris software on systems that match this rule. The rule also provides an example of rule wrap, which is defined in "Syntax of the `rules` File" on page 274.

`model` 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.

`network` The rule matches if the system is on subnet 192.168.2.0 and is an x86 based system. The `setup` begin script, the `x86_prof` profile, and the `done` finish script are used to install the Solaris software on systems that match the rule.

`memsize` The rule matches if the system has between 64 and 128 Mbytes of memory and is an x86 based system. The `prog_prof` profile is used to install the Solaris software on systems that match the rule.

`any` The 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 the rule. If `any` is used, it should always be the last rule in the `rules` file.

Creating a Profile

A profile is a text file that defines how to install the Solaris software on a system. A profile defines elements of the installation, for example, the software group to install. Every rule specifies a profile that defines how a system is to be installed. You can create different profiles for every rule or the same profile can be used in more than one rule.

A profile consists of one or more profile keywords and their values. Each profile keyword is a command that controls one aspect of how the JumpStart program is to install the Solaris software on a system. For example, the following profile keyword and value specify that the JumpStart program install the system as a server:

```
system_type  server
```

Note – If you created the JumpStart directory by using the procedures that are presented in "Creating a Profile Server for Networked Systems" on page 265 or "Creating a Profile Diskette for Standalone Systems" on page 269, sample profiles are already located in the JumpStart directory.

Syntax of Profiles

A profile must contain the following:

- The `install_type` profile keyword as the first entry
- One keyword per line
- The `root_device` keyword if the systems that are being upgraded by the profile contain more than one root (/) file system that can be upgraded

A profile can contain the following:

- Commented text

  Any text that is included after the # symbol on a line is treated by the JumpStart program as commented text. If a line begins with the # symbol, the entire line is treated as a comment.
- One or more blank lines

▼ To Create a Profile

1. **Use a text editor to create a text file. Name the file descriptively. Or, open a sample profile in the JumpStart directory that you created.**

   Note – Ensure that the name of the profile reflects how you intend to use the profile to install the Solaris software on a system. For example, you might name the profiles `basic_install`, `eng_profile`, or `user_profile`.

   ---

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

   For a list of profile keywords and values, see "Profile Keywords and Values" on page 340.

   Note – Profile keywords and their values are case sensitive.

   ---

3. **Save the profile in the JumpStart directory.**
4. **Ensure that `root` owns the profile and that the permissions are set to 644.**
5. **Test the profile (optional).**

   "Testing a Profile" on page 286 contains information about testing profiles.

Profile Examples

The following examples of profiles show how to use different profile keywords and profile values to control how the Solaris software is installed on a system. "Profile Keywords and Values" on page 340 contains a description of profile keywords and values.

EXAMPLE 26–2 Mounting Remote File Systems and Adding and Deleting Packages

```
# profile keywords        profile values
# -----------------       -----------------
  install_type            initial_install
  system_type             standalone
  partitioning            default
  filesys                 any 512 swap   # specify size of /swap
  cluster                 SUNWCprog
  package                 SUNWman delete
  cluster                 SUNWCacc
```

The following list describes some of the keywords and values from this example.

`install_type` The `install_type` keyword is required in every profile.

`system_type` The `system_type` keyword defines that the system is to be installed as a standalone system.

`partitioning` The file system slices are determined by the software to be installed with the value `default`. The size of `swap` is set to 512 Mbytes and is installed on any disk, value `any`.

`cluster` The Developer Solaris Software Group, `SUNWCprog`, is installed on the system.

`package` If the standard man pages are mounted from the file server, `s_ref`, on the network, the man page packages are not to be installed on the system. The packages that contain the System Accounting utilities are selected to be installed on the system.

EXAMPLE 26–3 Specifying Where to Install File Systems

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            initial_install
  system_type             standalone
  partitioning            explicit
  filesys                 c0t0d0s0 auto /
  filesys                 c0t3d0s1 auto swap
  filesys                 any auto usr
  cluster                 SUNWCall
```

The following list describes some of the keywords and values from this example.

EXAMPLE 26–3 Specifying Where to Install File Systems *(Continued)*

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

`cluster` The Entire Solaris Software Group, `SUNWCall`, is installed on the system.

EXAMPLE 26–4 x86: Using the `fdisk` Keyword

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

  fdisk                 c0t0d0 0x04 delete
  fdisk                 c0t0d0 solaris maxfree
  cluster               SUNWCall
  cluster               SUNWCacc delete
```

The following list describes some of the keywords and values from this example.

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

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

`cluster` The Entire Distribution software group, `SUNWCall`, is installed on the system.

`cluster` The system accounting utilities, `SUNWCacc`, are not to be installed on the system.

EXAMPLE 26–5 Reallocating Disk Space for an Upgrade

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            upgrade
  root_device             c0t3d0s2
  backup_media            remote_filesystem timber:/export/scratch
  layout_constraint       c0t3d0s2 changeable 100
  layout_constraint       c0t3d0s4 changeable
  layout_constraint       c0t3d0s5 movable
  package                 SUNWbcp delete
  package                 SUNWxwman add
  cluster                 SUNWCacc add
  locale                  de
```

EXAMPLE 26–5 Reallocating Disk Space for an Upgrade *(Continued)*

The following list describes some of the keywords and values from this example.

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

`root_device` — The root file system on `c0t3d0s2` is upgraded.

`backup_media` — A remote system that is named `timber` is to be used to back up data during the disk space reallocation. For more backup-media keyword values, see "`backup_media` Profile Keyword" on page 347.

`layout_constraint` — The `layout_constraint` keywords designate that auto-layout can perform the following when auto-layout attempts to reallocate disk space for the upgrade.

- Change slices 2 and 4. The slices can be moved to another location and the size can be changed.
- Move slice 5. The slice can be moved to another location but its size cannot change.

`package` — The binary compatibility package, `SUNWbcp`, is not installed on the system after the upgrade.

`package` — The code ensures that the X Window System man pages and the System Accounting Utilities are to be installed if they are not already installed on the system. All packages already on the system are automatically upgraded.

`locale` — The German localization packages are to be installed on the system.

EXAMPLE 26–6 Retrieving a Solaris Flash Archive From an HTTP Server

In the following example, the profile indicates that the custom JumpStart program retrieves the Solaris Flash archive from an HTTP server.

```
# profile keywords         profile values
# ----------------         -------------------
install_type               flash_install
archive_location           http://192.168.255.255/flasharchive/solarisarchive
partitioning               explicit
filesys                    c0t1d0s0 4000 /
filesys                    c0t1d0s1 512 swap
filesys                    c0t1d0s7 free /export/home
```

The following list describes some of the keywords and values from this example.

EXAMPLE 26–6 Retrieving a Solaris Flash Archive From an HTTP Server *(Continued)*

install_type The profile installs a Solaris Flash archive on the clone system. All files are overwritten as in an initial installation.

archive_location The Solaris Flash archive is retrieved from an HTTP server.

partitioning The file system slices are determined by the filesys keywords, value explicit. The size of root (/) is based on the size of the Solaris Flash archive. The root file system is installed on c0t1d0s0. The size of swap is set to the necessary size and is installed on c0t1d0s1. /export/home is based on the remaining disk space. /export/home is installed on c0t1d0s7.

EXAMPLE 26–7 Retrieving a Solaris Flash Archive From a Secure HTTP Server

In the following example, the profile indicates that the custom JumpStart program retrieves the Solaris Flash archive from a secure HTTP server.

```
# profile keywords         profile values
# ----------------         -------------------
install_type               flash_install
archive_location           https://192.168.255.255/solarisupdate.flar
partitioning               explicit
filesys                    c0t1d0s0 4000 /
filesys                    c0t1d0s1 512 swap
filesys                    c0t1d0s7 free /export/home
```

The following list describes some of the keywords and values from this example.

install_type The profile installs a Solaris Flash archive on the clone system. All files are overwritten as in an initial installation.

archive_location The compressed Solaris Flash archive is retrieved from a secure HTTP server.

partitioning The file system slices are determined by the filesys keywords, value explicit. The size of root (/) is based on the size of the Solaris Flash archive. The size of swap is set to the necessary size and is installed on c0t1d0s1. /export/home is based on the remaining disk space. /export/home is installed on c0t1d0s7.

EXAMPLE 26–8 Retrieving a Solaris Flash Differential Archive From an NFS Server

In the following example, the profile indicates that the custom JumpStart program retrieves the Solaris Flash archive from an NFS server. The flash_update keyword indicates that this is a differential archive. A differential archive installs only the differences between two system images.

EXAMPLE 26–8 Retrieving a Solaris Flash Differential Archive From an NFS Server *(Continued)*

```
# profile keywords         profile values
# ----------------         -------------------
install_type               flash_update
archive_location           nfs installserver:/export/solaris/flasharchive \
                           /solarisdiffarchive
no_master_check
```

The following list describes some of the keywords and values from this example.

`install_type` — The profile installs a Solaris Flash differential archive on the clone system. Only files that are specified by the archive are installed.

`archive_location` — The Solaris Flash archive is retrieved from an NFS server.

`no_master_check` — The clone system is not checked for a valid system image. A valid system image would have been built from the original master system.

EXAMPLE 26–9 Creating an Empty Boot Environment

In the following example, the profile indicates that the custom JumpStart program creates an empty boot environment. An empty boot environment contains no file systems and no copy from the current boot environment occurs. The boot environment can be populated later with a Solaris Flash archive and then activated.

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            initial_install
  system_type             standalone
  partitioning            explicit
  filesys                 c0t0d0s0 auto /
  filesys                 c0t3d0s1 auto swap
  filesys                 any auto usr
  cluster                 SUNWCall
  bootenv createbe bename second_BE \
  filesystem /:/dev/dsk/c0t1d0s0:ufs \
  filesystem -:/dev/dsk/c0t1d0s0:swap \
  filesystem /export:shared:ufs
```

The following list describes some of the keywords and values from this example.

`partitioning` — The file system slices are determined by the `filesys` keywords, value `explicit`. The size of root (/) is based on the selected software, value `auto`, and is installed on `c0t0d0s0`. The size of `swap` is set to the necessary size and is installed on `c0t3d0s1`. `usr` is based on the selected software and the installation program determines where `usr` is installed, based on the value `any`.

`cluster` The Entire Solaris Software Group, `SUNWCall`, is installed on the system.

`bootenv createbe` An empty, inactive boot environment is set up on disk `c0t1d0`. File systems for root (`/`), swap, and `/export` are created, but left empty. This second boot environment can be installed with a Solaris Flash archive at a later time. The new boot environment can then be activated to become the operating environment.

For keyword values and background on using this keyword, see the following references:

- For descriptions of keyword values, see "Profile Keywords and Values" on page 340.
- For background on using Solaris Live Upgrade that creates, upgrades, and activates inactive boot environments, see Chapter 33.
- For background on using a Solaris Flash archive, see Chapter 20.

EXAMPLE 26–10 Creating a RAID-1 Volume to Mirror the Root File System

In the following example, the profile indicates that the custom JumpStart program uses Solaris Volume Manager technology to create a RAID-1 volume (mirror) for the root (`/`) file system.

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            initial_install
  cluster                 SUNWCXall
  filesys                 mirror:d30 c0t1d0s0 c0t0d0s0  /
  filesys                 c0t0d0s3 512 swap
  metadb                  c0t0d0s4 size 8192 count 4
  metadb                  c0t1d0s4 size 8192 count 4
```

The following list describes some of the keywords and values from this example.

`cluster` The Entire Solaris Software Plus OEM Support software group, SUNWCXall, is installed on the system.

`filesys` The root (`/`) file system is created and mirrored on the slices `c0t1d0s0` and `c0t0d0s0`. The RAID-1 volume that mirrors `c0t1d0s0` and `c0t0d0s0` is named `d30`. The custom JumpStart program assigns names to the two submirrors.

`filesys` The `swap` file system is created and mirrored on the slice `c0t0d0s3`, and is sized at 512 Mbytes.

`metadb` Four state database replicas (metadbs) are installed on slice `c0t0d0s4`, and are sized at 8192 blocks (4 Mbytes.)

`metadb` Four state database replicas (metadbs) are installed on slice `c0t1d0s4`, and are sized at 8192 blocks (4 Mbytes.)

- For overview information about how to create mirrored file systems during your installation, see Chapter 10.
- For guidelines and requirements of creating mirrored file systems, see Chapter 11.
- For descriptions of keyword values, see "`filesys` Profile Keyword (Creating Mirrored File Systems)" on page 358 and "`metadb` Profile Keyword (Creating State Database Replicas)" on page 364.

EXAMPLE 26–11 Creating RAID-1 Volumes to Mirror Multiple File Systems

In the following example, the profile indicates that the custom JumpStart program uses Solaris Volume Manager technology to create RAID-1 volumes (mirrors) for the root (`/`), `swap`, and `/usr` file systems.

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            initial_install
  cluster                 SUNWCXall
  filesys                 mirror:d100 c0t1d0s0 c0t0d0s0 200 /
  filesys                 c0t1d0s5 500 /var
  filesys                 c0t0d0s5 500
  filesys                 mirror c0t0d0s1 512 swap
  metadb                  c0t0d0s3 size 8192 count 5
  filesys                 mirror c0t1d0s4 c0t0d0s4 2000 /usr
  filesys                 c0t1d0s7 free /export/home
  filesys                 c0t0d0s7 free
```

The following list describes some of the keywords and values from this example.

`cluster` The Entire Solaris Software Plus OEM Support software group, SUNWCXall, is installed on the system.

`filesys` The root (`/`) file system is created and mirrored on the slices `c0t1d0s0` and `c0t0d0s0`. The size of the root (`/`) file system is set to 200 Mbytes. The RAID-1 volume that mirrors `c0t1d0s0` and `c0t0d0s0` is named `d100`.

`filesys` The `/var` file system is installed on the slice `c0t1d0s5` and is sized at 500 Mbytes.The root (`/`) file system is created and mirrored on the slices `c0t1d0s0` and `c0t0d0s0`. The size of the root (`/`) file system is set to 200 Mbytes. The RAID-1 volume that mirrors `c0t1d0s0` and `c0t0d0s0` is named `d100`.

`filesys`	The `swap` file system is created and mirrored on the slice `c0t0d0s1`, and is sized at 512 Mbytes. The custom JumpStart program assigns a name to the mirror.
`metadb`	Five state database replicas (metadbs) are installed on slice `c0t0d0s3`, and are sized at 8192 blocks (4 Mbytes.)
`filesys`	The `/usr` file system is created and mirrored on the slices `c0t1d0s4` and `c0t0d0s4`. The size of the `/usr` file system is set to 2000 Mbytes. The custom JumpStart program assigns a name to the mirror.

- For overview information about how to create mirrored file systems during your installation, see Chapter 10.
- For guidelines and requirements of creating mirrored file systems, see Chapter 11.
- For descriptions of keyword values, see "`filesys` Profile Keyword (Creating Mirrored File Systems)" on page 358 and "`metadb` Profile Keyword (Creating State Database Replicas)" on page 364.

Testing a Profile

After you create a profile, use the `pfinstall`(1M) command to test the profile. Test the profile before you use the profile to install or upgrade a system. Testing a profile is especially useful when you are creating upgrade profiles that reallocate disk space.

By looking at the installation output that is generated by `pfinstall`, you can quickly determine if a profile works as you intended. For example, use the profile to determine if a system has enough disk space to upgrade to a new release of the Solaris software before you perform the upgrade on that system.

`pfinstall` enables you to test a profile against the following:

- The system's disk configuration where `pfinstall` is being run.
- Other disk configurations. You use a disk configuration file that represents a structure of a disk, for example, a disk's bytes/sector, flags, and slices. Creating disk configuration files is described in "Creating Disk Configuration Files" on page 302 and "x86: To Create a Disk Configuration File" on page 304.

Note – You cannot use a disk configuration file to test a profile you intend to use to upgrade a system. Instead, you must test the profile against the system's actual disk configuration and the software that is currently installed on that system.

▼ To Create a Temporary Solaris 9 Environment to Test a Profile

To test a profile for a particular Solaris release successfully and accurately, you must test a profile within the Solaris environment of the same release. For example, if you want to test a Solaris 9 initial installation profile, run the `pfinstall` command on a system that is running the Solaris 9 operating environment.

You need to create a temporary installation environment if you are testing a profile under one of the following conditions:

- You want to test a Solaris 9 upgrade profile on a system that is running a previous version of the Solaris software.
- You do not have a Solaris 9 system installed yet to test Solaris 9 initial installation profiles.

1. **Boot a system from an image of one of the following:**
   - Solaris *SPARC Platform Edition* DVD
   - Solaris *x86 Platform Edition* DVD
   - Solaris Software 1 of 2 *SPARC Platform Edition* CD
   - Solaris Software 1 of 2 *x86 Platform Edition* CD

   Note – If you want to test an upgrade profile, boot the system that you are upgrading.

   ---

2. **Respond to the system identification questions.**
3. **Exit from the installation program.**
   - If you are using the Solaris DVD, type ! at the following prompt.

     ```
     Solaris Web Start will assist you in installing software for Solaris.
     <Press ENTER to continue> {"!" exits}
     ```
   - If you are using the Solaris Software 1 of 2 CD, select the Solaris `suninstall` program as the program to install the Solaris 9 operating environment. On the first screen that is displayed, select Exit.
4. **Execute the `pfinstall` command from the shell. For details about using the `pfinstall` command, see Step 7 in "To Test a Profile" on page 288.**

▼ To Test a Profile

1. **Locate a system on which to test the profile that is the same type of platform, SPARC or x86, for which the profile was created.**

   If you are testing an upgrade profile, you must test the profile on the actual system that you intend to upgrade.

2. **Use the following decision table to determine what to do next.**

Test Scenario	Instructions
Test an initial installation profile and have a system that is running the Solaris 9 software	Become superuser on the system and go to Step 5.
Test an upgrade profile, or you do not have a system that is running Solaris 9 to test an initial installation profile	Create a temporary Solaris 9 environment to test the profile. For details, see "To Create a Temporary Solaris 9 Environment to Test a Profile" on page 287. Then, go to Step 3.

3. **Create a temporary mount point.**

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

4. **Mount the directory that contains the profile or profiles that you want to test.**

Mount Scenario	Typing Instructions
Mount a remote NFS file system for systems on the network	`mount -F nfs` *server_name*`:`*path* `/tmp/mnt`
SPARC: Mount a UFS-formatted diskette	`mount -F ufs /dev/diskette /tmp/mnt`
Mount a PCFS-formatted diskette	`mount -F pcfs /dev/diskette /tmp/mnt`

5. **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
```

6. **Did you mount a directory in Step 4?**

   - If yes, change the directory to `/tmp/mnt`.

     ```
     # cd /tmp/mnt
     ```

   - If no, change the directory to where the profile is located, which is usually the JumpStart directory.

     ```
     # cd jumpstart_dir_path
     ```

7. **Test the profile with the `pfinstall`(1M) command.**

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

Caution – You *must* include the `-d` or `-D` option. If you do not include one of these options, `pfinstall` uses the profile you specify to install the Solaris 9 software. All of the data on the system is overwritten.

`-D`

`pfinstall` uses the current system's disk configuration to test the profile. You must use the `-D` option to test an upgrade profile.

`-d` *disk_config_file*

`pfinstall` uses the disk configuration file, *disk_config_file*, to test the profile. If *disk_config_file* is not located in the directory where `pfinstall` is run, you must specify the path.

For instructions on how to create a disk configuration file, see "Creating Disk Configuration Files" on page 302.

Note – You cannot use the `-d` *disk_config_file* option with an upgrade profile, `install_type upgrade`. You must always test an upgrade profile against a system's disk configuration, that is, you must use the `-D` option.

`-c` *path*

The path to the Solaris 9 software image. You use this option, for example, if the system is using Volume Manager to mount the Solaris Software 1 of 2 CD for your platform.

Note – The `-c` option is not required if you booted from a Solaris DVD or a Solaris Software 1 of 2 CD image for your platform. The DVD or CD image is mounted on `/cdrom` as part of the booting process.

profile

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

Profile Test Examples

The following example shows how to use pfinstall to test a profile that is named basic_prof. The profile is tested against the disk configuration on a system on which the Solaris 9 software is installed. The basic_prof profile is located in the /jumpstart directory, and the path to the Solaris Software 1 of 2 *SPARC Platform Edition* CD or Solaris Software 1 of 2 *x86 Platform Edition* CD image is specified because Volume Manager is being used.

EXAMPLE 26–12 Profile Test Using a Solaris 9 System

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

The following example shows how to use pfinstall to test the profile that is named basic_prof on a Solaris 9 system. The test is performed against the 535_test disk configuration file. The test checks for 64 Mbytes of system memory. This example uses a Solaris Software 1 of 2 *SPARC Platform Edition* CD or Solaris Software 1 of 2 *x86 Platform Edition* CD image that is located in the /export/install directory.

EXAMPLE 26–13 Profile Test Using a Disk Configuration File

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

Validating the rules File

Before you can use a profile and rules file, you must run the check script to validate that the files are set up correctly. If all rules and profiles are correctly set up, the rules.ok file is created, which is required by the custom JumpStart installation software to match a system to a profile.

Table 26–3 describes what the check script does.

TABLE 26–3 What Happens When You Use the check Script

Stage	Description
1	The rules file is checked for syntax. check verifies that the rule keywords are legitimate and that the *begin*, *class*, and *finish* fields are specified for each rule. The *begin* and *finish* fields can consist of a minus sign (-) instead of a file name.

TABLE 26–3 What Happens When You Use the `check` Script *(Continued)*

Stage	Description
2	If no errors are found in the `rules` file, each profile that is specified in the rules is checked for syntax.
3	If no errors are found, `check` creates the `rules.ok` file from the `rules` file, removes all comments and blank lines, retains all rules, and adds the following comment line at the end: `# version=2 checksum=num`

▼ To Validate the `rules` File

1. **Ensure that the `check` script is located in the JumpStart directory.**

   Note – The `check` script is in the `Solaris_9/Misc/jumpstart_sample` directory on the Solaris DVD or on the Solaris Software 1 of 2 CD.

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

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

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

   `-p` *path* — Validates the `rules` by using the `check` script from the Solaris 9 software image instead of the `check` script from the system you are using. *path* is the image on a local disk or a mounted Solaris DVD or a Solaris Software 1 of 2 CD.

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

   `-r` *file_name* — Specifies a rules file other than the one that is named `rules`. Using this option, you can test the validity of a rule before you integrate the rule into the `rules` file.

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

4. **Ensure that `root` owns the `rules.ok` file and that the permissions are set to 644.**

After you validate the `rules` file, you can learn more about optional custom JumpStart features in Chapter 27. You can learn about performing custom JumpStart installations in Chapter 29.

CHAPTER 27

Using Optional Custom JumpStart Features (Tasks)

This chapter describes the optional features that are available to create additional custom JumpStart installation tools.

Note – Instructions in this chapter are valid for either a SPARC server or an x86 server that is being used to provide custom JumpStart files, called a profile server. A profile server can provide custom JumpStart files for different platform types. For example, a SPARC server can provide custom JumpStart files for both SPARC systems and x86 systems.

Creating Begin Scripts

A begin script is a user-defined Bourne shell script that you specify in the `rules` file. A begin script performs tasks before the Solaris software is installed on a system. You can use begin scripts only when using custom JumpStart to install the Solaris software.

Use a begin script to perform one of the following tasks:

- Create derived profiles
- Back up files before upgrading

Important Information About Begin Scripts

- 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 JumpStart program cannot mount the file systems onto /a, an error occurs and installation fails.
- During the installation, output from the begin script is deposited in /tmp/begin.log. After the installation is completed, the log file is redirected to /var/sadm/system/logs/begin.log.
- Ensure that root owns the begin script and that the permissions are set to 644.
- You can use custom JumpStart Environment variables in your begin scripts. For a list of environment variables, see "Custom JumpStart Environment Variables" on page 369.
- Save begin scripts in the JumpStart directory.

Creating Derived Profiles With a Begin Script

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. For example, you might need to use derived profiles for identical system models that have different hardware components, such as systems that contain different frame buffers.

To set up a rule to use a derived profile, you must perform the following tasks:

- Set the profile field to an equal sign (=) instead of a profile.
- Set the begin field to a begin script that creates a derived profile that depends on the system on which you intend to install Solaris.

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.

The following is an example of a begin script that creates the same derived profile every time. You can write a begin script to create different derived profiles that depend on the evaluation of rules.

EXAMPLE 27–1 Begin Script That Creates a Derived Profile

```
#!/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}
```

In the example, 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, ensure the script does not have any errors. A derived profile is not verified by the check script because derived profiles are not created until the execution of the begin script.

Creating Finish Scripts

A finish script is a user-defined Bourne shell script that you specify in the rules file. A finish script performs tasks after the Solaris software is installed on a system, but before the system reboots. You can use finish scripts only when using custom JumpStart to install Solaris.

Tasks that you can perform with a finish script include the following:

- Add files
- Add individual packages or patches in addition to the ones that are installed in a particular software group
- Customize the root environment
- Set the system's root password
- Install additional software

Important Information About Finish Scripts

- The Solaris suninstall program mounts the system's file systems on /a. The file systems remain mounted on /a until the system reboots. You can use the finish script to add, change, or remove files from the newly installed file system hierarchy by modifying the file systems that are respective to /a.
- During the installation, output from the finish script is deposited in /tmp/finish.log. After the installation is completed, the log file is redirected to /var/sadm/system/logs/finish.log.
- Ensure that root owns the finish script and that the permissions are set to 644.
- You can use custom JumpStart Environment variables in your finish scripts. For a list of environment variables, see "Custom JumpStart Environment Variables" on page 369.

- Save finish scripts in the JumpStart directory.

▼ To Add Files With a Finish Script

Through a finish script, you can add files from the JumpStart directory to an already installed system. You can add the files because the JumpStart directory is mounted on the directory that is specified by the `SI_CONFIG_DIR` variable. The directory 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.

---

1. **Copy all of the files that you are adding to the installed system to the JumpStart directory.**
2. **Insert the following line in the finish script for each file that you want to be copied to the newly installed file system hierarchy:**

   ```
   cp ${SI_CONFIG_DIR}/file_name  /a/path_name
   ```

For example, assume you have a special application, `site_prog`, developed for all users at your site. If you place a copy of `site_prog` into the JumpStart directory, the following line in a finish script copies `site_prog` from the JumpStart directory into a system's `/usr/bin` directory:

```
cp ${SI_CONFIG_DIR}/site_prog  /a/usr/bin
```

Adding Packages or Patches With a Finish Script

You can create a finish script to automatically add packages or patches after the Solaris software is installed on a system. By adding packages with a finish script, you reduce time and ensure consistency in what packages and patches are installed on different systems at your site.

When you use the `pkgadd`(1M) or `patchadd`(1M) commands in finish scripts, use the `-R` option to specify `/a` as the root path.

- Example 27–2 shows an example of a finish script that adds packages.
- Example 27–3 shows an example of a finish script that adds patches.

EXAMPLE 27–2 Adding Packages With a Finish Script

```
#!/bin/sh

BASE=/a
```

EXAMPLE 27–2 Adding Packages With a Finish Script *(Continued)*

```
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin

mkdir ${MNT}
mount -f nfs sherlock:/export/package ${MNT}
cat >${ADMIN_FILE} <<DONT_ASK
mail=root
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=ask
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
DONT_ASK

/usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNWxyz
umount ${MNT}
rmdir ${MNT}
```

The following describes some commands for this example.

- The following command mounts a directory on a server that contains the package to install.

  ```
  mount -f nfs sherlock:/export/package ${MNT}
  ```

- The following command creates a temporary package administration file, `admin`, to force the `pkgadd`(1M) command not to perform checks or prompt for questions when installing a package. Use the temporary package administration file to maintain a hands-off installation when you are adding packages.

  ```
  cat >${ADMIN_FILE} <<DONT_ASK
  ```

- The `pkgadd` command below adds the package by using the `-a` option, specifying the package administration file, and the `-R` option, specifying the root path.

  ```
  /usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNWxyz
  ```

EXAMPLE 27–3 Adding Patches With a Finish Script

```
 #!/bin/sh

########
#
# USER-CONFIGURABLE OPTIONS
#
########

# The location of the patches to add to the system after it's installed.
```

EXAMPLE 27–3 Adding Patches With a Finish Script *(Continued)*

```
# The OS rev (5.x) and the architecture (‘mach‘) will be added to the
# root.  For example, /foo on a 8 SPARC would turn into /foo/5.8/sparc
LUPATCHHOST=ins3525-svr
LUPATCHPATHROOT=/export/solaris/patchdb
#########
#
# NO USER-SERVICEABLE PARTS PAST THIS POINT
#
#########

BASEDIR=/a

# Figure out the source and target OS versions
echo Determining OS revisions...
SRCREV=‘uname -r‘
echo Source $SRCREV

LUPATCHPATH=$LUPATCHPATHROOT/$SRCREV/‘mach‘

#
# Add the patches needed
#
echo Adding OS patches
mount $LUPATCHHOST:$LUPATCHPATH /mnt >/dev/null 2>&1
if [ $? = 0 ] ; then
    for patch in ‘cat /mnt/*Recommended/patch_order‘ ; do
        (cd /mnt/*Recommended/$patch ; echo yes | patchadd -u -d -R $BASEDIR .)
    done
    cd /tmp
    umount /mnt
else
    echo "No patches found"
if
```

Note – In the past, the `chroot`(1M) command was used with the `pkgadd` and `patchadd` commands in the finish script environment. In rare instances, some packages or patches do not work with the `-R` option. You must create a dummy `/etc/mnttab` file in the `/a` root path before issuing the `chroot` command.

To create a dummy `/etc/mnttab` file, add the following line to your finish script:

```
cp /etc/mnttab /a/etc/mnttab
```

Customizing the Root Environment With a Finish Script

You can also use finish scripts to customize files that are already installed on a system. For example, the finish script in Example 27–4 customizes the root environment by appending information to the `.cshrc` file in the root (/) directory.

EXAMPLE 27–4 Customizing the Root Environment With a Finish Script

```
#!/bin/sh
#
# Customize root’s environment
#
echo "***adding customizations in /.cshrc"
test -f a/.cshrc || {
cat >> a/.cshrc <<EOF
set history=100 savehist=200 filec ignoreeof prompt="\$user@`uname -n`> "
alias cp cp -i
alias mv mv -i
alias rm rm -i
alias ls ls -FC
alias h history
alias c clear
unset autologout
EOF
}
```

Setting a System’s Root Password With a Finish Script

After the Solaris software is installed on a system, the system reboots. Before the boot process is completed, the system prompts for the root password. Until someone types a password, the system cannot finish booting.

A finish script that is named `set_root_pw` is saved in the `auto_install_sample` directory. The finish script shows how to set the root password automatically, without prompting. `set_root_pw` is shown in Example 27–5.

Note – If you set the system’s root password with a finish script, users might attempt to discover the root password from the encrypted password in your finish script. Ensure that you safeguard against users who might try to determine the root password.

EXAMPLE 27–5 Setting the System’s Root Password With a Finish Script

```
#!/bin/sh
#
#       @(#)set_root_pw 1.4 93/12/23 SMI
```

EXAMPLE 27–5 Setting the System's Root Password With a Finish Script *(Continued)*

```
    #
    # 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
PASSWD=dKO5IBkSF42lw
    #create a temporary input file
cp /a/etc/shadow /a/etc/shadow.orig

    mv /a/etc/shadow /a/etc/shadow.orig
    nawk -F: '{
        if ( $1 == "root" )
          printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,passwd,$3,$4,$5,$6,$7,$8,$9
     else
               printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,$2,$3,$4,$5,$6,$7,$8,$9
     }' passwd="$PASSWD" /a/etc/shadow.orig > /a/etc/shadow
#remove the temporary file
rm -f /a/etc/shadow.orig
# 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}
```

The following describes some of the commands in this example.

- The following command sets the variable `PASSWD` to an encrypted root password that is obtained from an existing entry in a system's `/etc/shadow` file.

  ```
  #create a temporary input file
  ```

- The following command creates a temporary input file of `/a/etc/shadow`.

  ```
  cp /a/etc/shadow /a/etc/shadow.orig
  ```

- The following command changes the root entry in the `/etc/shadow` file for the newly installed system by using `$PASSWD` as the password field.

  ```
  if ( $1 == "root" )
  ```

- The following command removes the temporary `/a/etc/shadow` file.

  ```
  rm -f /a/etc/shadow.orig
  ```

- The following command changes the entry from `0` to a `1` in the state file so that the user is not prompted for the root password. The state file is accessed by using the variable `SI_SYS_STATE`, which has a value currently of `/a/etc/.sysIDtool.state`. To avoid problems with your scripts if this value changes, always reference this file by using `$SI_SYS_STATE`. The `sed` command that is shown here contains a tab character after the `0` and after the `1`.

  ```
  sed -e 's/0 # root/1 # root/' ${SI_SYS_STATE} > /tmp/state.$$
  ```

Installing Software With Web Start Installation Programs With Finish Scripts

You can use finish scripts to install additional software after the Solaris operating environment is installed. Some software programs are installed by the Solaris Web Start program, which prompts you to enter information during the installation. To maintain a hands-off installation, you can run the Solaris Web Start program with the `-nodisplay` or `-noconsole` options.

TABLE 27–1 Solaris Web Start Options

Option	Description
`-nodisplay`	Runs the installer without a graphic user interface. Use the default product installation unless the installation was modified by the `-locales` option.
`-noconsole`	Runs the installation without any interactive text console device. Useful when paired with `-nodisplay` for UNIX script use.

For more information, see the man page `installer`(1M).

Creating a Compressed Configuration File

Rather than using the `add_install_client` command to specify the location of the custom JumpStart configuration files, you can specify the location of the files when you boot the system. However, you can only specify the name of one file when you issue the `boot` command. As a result, you must compress all of the custom JumpStart configuration files into one file. The compressed configuration file can be one of the following types:

- `tar`
- Compressed `tar`
- `zip`
- `bzip tar`

▼ To Create a Compressed Configuration File

1. **Change the directory to the JumpStart directory on the profile server.**

   ```
   # cd jumpstart_dir_path
   ```

2. **Use a compression tool to compress the custom JumpStart configuration files into one file.**

   Note – The compressed configuration file cannot contain relative paths. The custom JumpStart configuration files must be in the same directory as the compressed file.

   The compressed configuration file must contain the following files:

   - Profile
   - rules
   - rules.ok

   You can also include the sysidcfg file in the compressed configuration file.

3. **Save the compressed configuration file on an NFS server, an HTTP server, or on a local hard disk.**

Compressed Configuration File Example

The following example shows how to use the tar command to create a compressed configuration file that is named config.tar. The custom JumpStart configuration files are located in the /jumpstart directory.

EXAMPLE 27–6 Creating a Compressed Configuration File

```
# cd /jumpstart
# tar -cvf config.tar *
a profile 1K
a rules 1K
a rules.ok 1K
a sysidcfg 1K
```

Creating Disk Configuration Files

This section describes how to create single-disk and multiple-disk configuration files. Disk configuration files enable you to use pfinstall(1M) from a single system to test profiles against different disk configurations.

▼ SPARC: To Create a Disk Configuration File

1. **Locate a SPARC system with a disk you want to test.**

2. **Become superuser.**

3. **Create a single–disk configuration file by redirecting the output of the `prtvtoc`(1M) command to a file.**

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

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

   disk_config_file — The name of the disk configuration file.

4. **Determine if you are testing the installation of Solaris software on multiple disks.**

   - If no, stop. You are finished.
   - If yes, concatenate the single–disk configuration files and save the output in a new file.

     ```
     # cat  disk_file1 disk_file2  >multi_disk_config
     ```

     The new file becomes the multiple-disk configuration file, as in the following example.

     ```
     # cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
     ```

5. **Determine if the target numbers in the disk device names are unique in the multiple-disk configuration file that you created in the previous step.**

   - If yes, stop. You are finished.
   - If no, open the file with a text editor and make the target numbers unique in the disk device names.

     For example, assume that the file contains the same target number, `t0`, for different disk device names, as shown here.

     ```
     * /dev/rdsk/c0t0d0s2 partition map
     ...
     * /dev/rdsk/c0t0d0s2 partition map
     ```

     Change the second target number to `t2`, as shown here:

     ```
     * /dev/rdsk/c0t0d0s2 partition map
     ...
     * /dev/rdsk/c0t2d0s2 partition map
     ```

SPARC: Disk Configuration File Example

The following example shows how to create a single–disk configuration file, `104_test`, on a SPARC system with a 104-Mbyte disk.

EXAMPLE 27–7 SPARC: Creating a Disk Configuration File

You redirect the output of the `prtvtoc` command to a single–disk configuration file that is named `104_test`:

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

The contents of the `104_test` file resemble the following:

```
* /dev/rdsk/c0t3d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      72 sectors/track
*      14 tracks/cylinder
*    1008 sectors/cylinder
*    2038 cylinders*    2036 accessible cylinders
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       1      2    00           0     164304   164303   /
       2      5    00           0    2052288  2052287
       3      0    00      164304     823536   987839   /disk2/b298
       5      0    00      987840     614880  1602719   /install/298/sparc/work
       7      0    00     1602720     449568  2052287   /space
```

You have created disk configuration files for a SPARC based system. "Testing a Profile" on page 286 contains information about using disk configuration files to test profiles.

▼ x86: To Create a Disk Configuration File

1. **Locate an x86 based system that contains a disk that you are testing.**

2. **Become superuser.**

3. **Create part of the single-disk configuration file by saving the output of the `fdisk`(1M) command in a file.**

   ```
   # fdisk -R -W disk_config_file -h /dev/rdsk/device_name
   ```

   disk_config_file   The name of a disk configuration file.

/dev/rdsk/*device_name* The device name of the fdisk layout of the entire disk. *device_name* must be in the form c*w*t*x*d*y*p0 or c*x*d*y*p0.

4. **Append the output of the prtvtoc(1M) command to the disk configuration file:**

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

/dev/rdsk/*device_name* The device name of the system's disk. *device_name* must be in the form c*w*t*x*d*y*s2 or c*x*d*y*s2.

disk_config The name of the disk configuration file

5. **Determine if you are testing the installation of Solaris software on multiple disks.**

- If no, stop. You are finished.
- If yes, concatenate the single–disk configuration files and save the output in a new file.

```
# cat disk_file1 disk_file2 >multi_disk_config
```

The new file becomes the multiple-disk configuration file, as in the following example.

```
# cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
```

6. **Determine if the target numbers in the disk device names are unique in the multiple-disk configuration file that you created in the previous step.**

- If yes, stop. You are finished.
- If no, open the file with a text editor and make the target numbers unique.

For example, if the file contains the same target number, t0, for different disk device names as shown here:

```
* /dev/rdsk/c0t0d0s2 partition map
...
* /dev/rdsk/c0t0d0s2 partition map
```

Change the second target number to t2, as shown here:

```
* /dev/rdsk/c0t0d0s2 partition map
...
* /dev/rdsk/c0t2d0s2 partition map
```

x86: Disk Configuration File Example

The following example shows how to create a single–disk configuration file, 500_test, on an x86 system that contains a 500-Mbyte disk.

EXAMPLE 27–8 x86: Creating a Disk Configuration File

First, you save the output of the `fdisk` command to a file that is named `500_test`:

```
# fdisk -R -W 500_test -h /dev/rdsk/c0t0d0p0
```

The `500_test` file looks like the following:

```
 * /dev/rdsk/c0t0d0p0 default fdisk table
* Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
*  HBA Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
* systid:
*  1:    DOSOS12
*  2:    PCIXOS
*  4:    DOSOS16
*  5:    EXTDOS
*  6:    DOSBIG
*  86:   DOSDATA
*  98:   OTHEROS
*  99:   UNIXOS
* 130:   SUNIXOS
*
* Id  Act Bhead Bsect   Bcyl  Ehead  Esect  Ecyl Rsect  Numsect
 130  128 44    3       0     46    30     1001 1410   2050140
```

Second, you append the output of the `prtvtoc` command to the `500_test` file:

```
# prtvtoc /dev/rdsk/c0t0d0s2 >>500_test
```

The `500_test` file is now a complete disk configuration file:

```
* /dev/rdsk/c0t0d0p0 default fdisk table
* Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
*  HBA Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
* systid:
```

EXAMPLE 27–8 x86: Creating a Disk Configuration File *(Continued)*

```
*  1:    DOSOS12
*  2:    PCIXOS
*  4:    DOSOS16
*  5:    EXTDOS
*  6:    DOSBIG
*  86:   DOSDATA
*  98:   OTHEROS
*  99:   UNIXOS
*  130:  SUNIXOS
*
* Id  Act Bhead Bsect Bcyl  Ehead  Esec  Ecyl Rsect  Numsect
 130  128 44    3     0     46    30    1001 1410   2050140
* /dev/rdsk/c0t0d0s2 partition map
*
* Dimensions:
*      512 bytes/sector
*       94 sectors/track
*       15 tracks/cylinder
*     1110 sectors/cylinder
*     1454 cylinders
*     1452 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*                          First    Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       2      5    01       1410   2045910   2047319
       7      6    00       4230   2043090   2047319  /space
       8      1    01          0      1410      1409
       9      9    01       1410      2820     422987
```

You have created disk configuration files for an x86 based system. "Testing a Profile" on page 286 contains information about using disk configuration files to test profiles.

Using a Site-Specific Installation Program

You can also use begin and finish scripts to create your own installation program to install Solaris software.

When you specify a minus sign (`-`) in the profile field, begin and finish scripts control how Solaris software is installed on a system instead of the profile and the Solaris `suninstall` program.

For example, if the following rule matches a system, the x_install.beg begin script and the x_install.fin finish script install Solaris software on the system that is named clover:

```
hostname clover x_install.beg - x_install.fin
```

CHAPTER 28

Creating Custom Rule and Probe Keywords (Tasks)

This chapter provides information and procedures for creating your own custom rule and probe keywords.

- "Probe Keywords" on page 309
- "Creating a custom_probes File" on page 310
- "Validating the custom_probes File" on page 313

Probe Keywords

To understand what a probe keyword is, you first need to recall what a rule keyword is. A rule keyword is a predefined lexical unit or word that describes a general system attribute, such as host name, hostname, or memory size, memsize. Rule keywords and the values that are associated with them enable you to match a system that has the same attribute to a profile. This match of a system's attributes defines how the Solaris software is to be installed on each system in the group.

Custom JumpStart environment variables, which you use in begin and finish scripts, are set on demand. For example, information about which operating system is already installed on a system is only available in SI_INSTALLED after the installed rule keyword is used.

In some situations, you might need to extract the same information in a begin or finish script for a purpose other than to match a system and run a profile. Probe keywords provide the solution. Probe keywords extract attribute information and remove the need for you to set up a matching condition and run a profile.

For a list of probe keywords and values, see "Probe Keywords and Values" on page 372.

Creating a custom_probes File

If the rule and probe keywords that are described in "Rule Keywords and Values" on page 335 and "Probe Keywords and Values" on page 372 are not precise enough for your needs, you can define your own custom rule or probe keywords by creating a custom_probes file.

The custom_probes file is a Bourne shell script that contains two types of functions. You must save the custom_probes file in the same JumpStart directory where you saved the rules file. The two types of functions that you can define in a custom_probes file are as follows:

- Probe – Gathers the information you want or does the actual work and sets a corresponding SI_ environment variable that you define. Probe functions become probe keywords.
- Comparison – Calls a corresponding probe function, compares the output of the probe function, and returns 0 if the keyword matches or 1 if the keyword does not match. Comparison functions become rule keywords.

Syntax of the custom_probes File

The custom_probes file can contain any valid Bourne shell command, variable, or algorithm.

Note – You can define probe and comparison functions that require a single argument in the custom_probes file. When you use the corresponding custom probe keyword in the rules file, the argument after the keyword is interpreted (as $1).

When you use the corresponding custom rule keyword in the rules file, the argument is interpreted as starting after the keyword and ending before the next && or begin script, whichever comes first.

The custom_probes file must meet the following requirements:

- Have the name custom_probes
- Have root as its owner
- Be executable and have permissions set to 755
- Contain at least one probe function and one corresponding comparison function

To improve clarity and organization, define all probe functions first, at the top of the file, followed by all comparison functions.

Syntax of Function Names in custom_probes

The name of a probe function must begin with probe_. The name of a comparison function must begin with cmp_.

Functions that begin with probe_ define new probe keywords. For example, the function probe_tcx defines the new probe keyword tcx. Functions that begin with cmp_ define new rule keywords. For example, cmp_tcx defines the new rule keyword tcx.

▼ To Create a custom_probes File

1. **Use a text editor to create a Bourne shell script text file. Name the file custom_probes.**

2. **In the custom_probes text file, define your probe and comparison functions.**

   Note – You can define probe and comparison functions that require arguments in the custom_probes file. When you use the corresponding custom probe keyword in the rules file, the arguments after the keyword are interpreted in sequence (as $1, $2, and so on).

   When you use the corresponding custom rule keyword in the rules file, the arguments are interpreted in sequence after the keyword and before the next && or begin script, whichever comes first.

3. **Save the custom_probes file in the JumpStart directory next to the rules file.**

4. **Ensure that root owns the rules file and that the permissions are set to 644.**

Examples of a custom_probes File and Keyword

You can find additional examples of probe and comparison functions in the following directories:

- /usr/sbin/install.d/chkprobe on a system that has the Solaris software installed
- /Solaris_9/Tools/Boot/usr/sbin/install.d/chkprobe on the Solaris DVD or on the Solaris Software 1 of 2 CD

The following custom_probes file contains a probe and comparison function that tests for the presence of a TCX graphics card.

EXAMPLE 28–1 `custom_probes` File

```
#!/bin/sh
#
# custom_probe script to test for the presence of a TCX graphics card.
#

#
# PROBE FUNCTIONS
#
probe_tcx() {
  SI_TCX=‘modinfo | grep tcx | nawk ’{print $6}’‘
  export SI_TCX
}

#
# COMPARISON FUNCTIONS
#
cmp_tcx() {
  probe_tcx

  if [ "X${SI_TCX}" = "X${1}" ]; then
     return 0
  else
     return 1
  fi
}
```

The following example `rules` file shows the use of the probe keyword that is defined in the preceding example, `tcx`. If a TCX graphics card is installed and found in a system, `profile_tcx` is run. Otherwise, `profile` is run.

Note – Always place probe keywords at or near the beginning of the `rules` file to ensure that the keywords are read and run before other rule keywords that might rely on the probe keywords.

EXAMPLE 28–2 Custom Probe Keyword Used in a `rules` File

```
probe tcx
tcx     tcx     -     profile_tcx     -
any     any     -     profile         -
```

Validating the custom_probes File

Before you can use a profile, rules, and custom_probes file, you must run the check script to validate that the files are set up correctly. If all profiles, rules, and probe and comparison functions are correctly set up, the rules.ok and custom_probes.ok files are created. Table 28–1 describes what the check script does.

TABLE 28–1 What Happens When You Use the check Script

Stage	Description
1	check searches for a custom_probes file.
2	If the file exists, check creates the custom_probes.ok file from the custom_probes file, removes all comments and blank lines, and retains all Bourne shell commands, variables, and algorithms. Then, check adds the following comment line at the end: # version=2 checksum=num

▼ To Validate the custom_probes File

1. **Verify that the check script is located in the JumpStart directory.**

   Note – The check script is in the Solaris_9/Misc/jumpstart_sample directory on the Solaris DVD or on the Solaris Software 1 of 2 CD.

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

3. **Run the check script to validate the rules and custom_probes files.**

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

-p *path* Validates the custom_probes file by using the check script from the Solaris 9 software image for your platform instead of the check script from the system you are using. *path* is the image on a local disk or a mounted Solaris DVD or Solaris Software 1 of 2 CD.

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

-r *file_name* Specifies a file name other than the one that is named custom_probes. By using the -r option, you can test the validity of a set of functions before integrating the functions into the custom_probes file.

As the check script runs, the script reports the validity of the rules and custom_probes files and each profile. If no errors are encountered, the script reports: “The custom JumpStart configuration is ok” and creates the rules.ok and custom_probes.ok files in the JumpStart directory.

4. **Determine if the custom_probes.ok file is executable.**
   - If yes, go to Step 5.
   - If no, type the following command.

     ```
     # chmod +x custom_probes
     ```

5. **Ensure that root owns the custom_probes.ok file and that the permissions are set to 755.**

CHAPTER **29**

Performing a Custom JumpStart Installation (Tasks)

This chapter describes how to perform a custom JumpStart installation on a SPARC based or an x86 based system. You need to follow these procedures on the system on which you intend to install the Solaris 9 software.

- "SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program" on page 318
- "x86: To Perform an Installation or Upgrade With the Custom JumpStart Program" on page 322

SPARC: Performing a Custom JumpStart Installation

SPARC: Task Map: Setting Up a System for a Custom JumpStart Installation

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

TABLE 29–1 Task Map: Setting Up a System for a Custom JumpStart Installation

Task	Description	For Instructions
Check if the system is supported	Check the hardware documentation for system support in the Solaris 9 environment.	*Solaris 9 Sun Hardware Platform Guide*
Check if the system has enough disk space for the Solaris 9 software	Verify that you have planned enough space to install the Solaris software on your system.	Chapter 5
(Optional) Preconfigure system configuration information	You can use the `sysidcfg` file or the name service to preconfigure installation information for a system. If you preconfigure system information, the installation program does not prompt you to supply the information during the installation.	Chapter 7
Prepare the system for custom JumpStart installation	Create and validate a `rules` file and profile files.	Chapter 26
(Optional) Prepare optional custom JumpStart features	If you are using begin scripts, finish scripts, or other optional features, prepare the scripts or files.	Chapter 27 and Chapter 28
(Optional) Set up the system to install over the network	To install a system from a remote Solaris DVD or Solaris Software *SPARC Platform Edition* CD image, you need to set up the system to boot and install from an install server or a boot server.	Chapter 15
(Optional) Prepare for a Solaris Flash archive installation	Before installing a Solaris Flash archive, you need to follow specific steps.	"To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation" on page 317
Install or upgrade	Boot the system to initiate the installation or upgrade.	"SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program" on page 318

▼ To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation

You can install a full archive for an initial installation or a differential archive for an update. You can use the custom JumpStart installation method or use Solaris Live Upgrade to install an archive on an inactive boot environment.

- For an overview of a full or differential archive, see Chapter 20.
- For procedures about installing an archive on an inactive boot environment by using Solaris Live Upgrade, see "To Install a Solaris Flash Archive With a Profile (Command-Line Interface)" on page 459.

1. **On the install server, create the custom JumpStart `rules` file.**

   For detailed instructions about creating custom JumpStart files, refer to Chapter 26.

2. **On the install server, create the custom JumpStart profile file.**

   For examples of Solaris Flash archive profiles, see "Profile Examples" on page 279.

   From the existing list of custom JumpStart keywords in Table 31–2, the only keywords valid when you install a Solaris Flash archive are the following:

TABLE 29–2 Solaris Flash Archive Keywords

Keyword	Initial Installation	Differential Archive
(required) `archive_location`	X	X
`fdisk` (x86 only)	X	X
`filesys` **Note –** You cannot set the `filesys` keyword to the value `auto`.	X	
`forced_deployment`		X
(required) `install_type`	X	X
`local_customization`	X	X
`no_content_check`		X
`no_master_check`		X
`root_device`	X	X

   a. **Set the value of the keyword `install_type` to one of the following types.**

   - For a full archive installation, set the value to `flash_install`
   - For a differential archive installation, set the value to `flash_update`

b. **Add the path to the Solaris Flash archive by using the `archive_location` keyword.**

   For details about the `archive_location` keyword, refer to "`archive_location` Keyword" on page 342.

c. **Specify the file system configuration.**

   The Solaris Flash archive extraction process does not support auto-layout of partitions.

d. **(Optional) If you want to install an additional Solaris Flash archive on the clone system, add one `archive_location` line for each archive that you want to install.**

3. **On the install server, add the clients that you are installing with the Solaris Flash archive.**

   For detailed instructions, refer to "Adding Systems to Be Installed From the Network With a DVD Image" on page 137 or "Adding Systems to Be Installed From the Network With a CD Image" on page 167.

4. **Perform the custom JumpStart installation on the clone systems.**

   For detailed instructions, refer to "SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program" on page 318.

▼ SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program

1. **If the system is part of a network, ensure that an Ethernet connector or similar network adapter is attached to your system.**

2. **If you are installing a system that is connected through a `tip`(1) line, ensure that your window display is at least 80 columns wide and 24 rows long.**

   To determine the current dimensions of your `tip` window, use the `stty`(1) command.

3. **If you are using the system's DVD-ROM or CD-ROM drive to install the Solaris 9 software, insert the Solaris *SPARC Platform Edition* DVD or the Solaris Software 1 of 2 *SPARC Platform Edition* CD in the drive.**

4. **If you are using a profile diskette, insert the profile diskette in the system's diskette drive.**

5. **Boot the system.**

   - If the system is new, out–of–the–box, turn on the system.
   - If you want to install or upgrade an existing system, shut down the system. At the `ok` prompt, type the following command.

```
ok boot cdrom:net - install [url:ask] [dhcp] [nowin]
```

cdrom
: Specifies to boot from a CD or a DVD.

  For a system with an older EEPROM, replace `cdrom` with `sd(0,6,2)` to boot from the system's CD-ROM or DVD-ROM drive.

net
: Specifies to boot from an install server on the network.

url
: Specifies the location of the custom JumpStart files. You can specify a URL for files that are located in the following places:

  - Local hard disk

    `file://`*jumpstart_dir_path/compressed_config_file*

  - NFS server

    `nfs://`*server_name:IP_address/jumpstart_dir_path/compressed_config_file*

  - HTTP server

    `http://`*server_name:IP_address/jumpstart_dir_path/compressed_config_file&proxy_info*

  If you placed a `sysidcfg` file in the compressed configuration file, you must specify the IP address of the server that contains the file, as in the following example:

  ```
  http://131.141.2.32/jumpstart/config.tar
  ```

  If you saved the compressed configuration file on an HTTP server that is behind a firewall, you must use a proxy specifier during boot. You do not need to specify an IP address for the server that contains the file. You must specify an IP address for the proxy server, as in the following example:

  ```
  http://www.shadow.com/jumpstart/
  config.tar&proxy=131.141.6.151
  ```

ask
: Specifies that the installation program prompt you to type the location of the compressed configuration file after the system boots and connects to the network.

  If you bypass the prompt by pressing Return, the installation program interactively configures the network parameters. The installation program then prompts you for the location of the compressed configuration file. If you bypass the prompt by pressing Return, the Solaris `suninstall` program begins.

dhcp
: Specifies to use a DHCP server to obtain network installation information that is needed to boot the system.

  If you do not specify to use a DHCP server, the system uses the `/etc/bootparams` file or the name service `bootparams` database.

nowin Specifies not to begin the X program. You do not need to use the X program to perform a custom JumpStart installation, so you can reduce the installation time by using the `nowin` option.

SPARC only – The system checks hardware and system components and your SPARC system boots. Booting lasts several minutes.

---

6. **If you did not preconfigure system information in the `sysidcfg` file, when prompted, answer the questions about system configuration.**
7. **Follow the instructions on the screen to install the software.**

   When the JumpStart program finishes installing the Solaris software, the system reboots automatically.

   After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

   - `/var/sadm/system/logs`
   - `/var/sadm/install/logs`

x86: Performing a Custom JumpStart Installation

x86: Task Map: Setting Up a System for a Custom JumpStart Installation

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

TABLE 29–3 x86: Task Map: Setting Up a System for a Custom JumpStart Installation

Task	Description	For Instructions
Determine if you need to preserve an existing operating system and user data	If the existing operating system on the system uses the entire disk, you must preserve the existing operating system so it can co-exist with the Solaris 9 software. This decision determines how to specify the `fdisk`(1M) keyword in the system's profile.	"x86: `fdisk` Profile Keyword" on page 353
Check if the system is supported	Check the hardware documentation for system support in the Solaris 9 environment.	Hardware manufacturer's documentation
Check if the system has enough disk space for the Solaris 9 software	Verify that you have planned enough space to install the Solaris software on your system.	Chapter 5
(Optional) Preconfigure system configuration information	You can use the `sysidcfg` file or the name service to preconfigure installation information for a system. If you preconfigure system information, the installation program does not prompt you to supply the information during the installation.	Chapter 7
Prepare the system for custom JumpStart installation	Create and validate a `rules` file and profile files.	Chapter 26
(Optional) Prepare optional custom JumpStart features	If you are using `begin` scripts, `finish` scripts, or other optional features, prepare the scripts or files.	Chapter 27 and Chapter 28
(Optional) Set up the system to install over the network	To install a system from a remote Solaris *x86 Platform Edition* DVD or Solaris Software *x86 Platform Edition* CD image, you need to set up the system to boot and install from an install server or a boot server.	Chapter 15

TABLE 29–3 x86: Task Map: Setting Up a System for a Custom JumpStart Installation *(Continued)*

Task	Description	For Instructions
Install or upgrade	Boot the system to initiate the installation or upgrade.	"x86: To Perform an Installation or Upgrade With the Custom JumpStart Program" on page 322

▼ x86: To Perform an Installation or Upgrade With the Custom JumpStart Program

1. **If the system is part of a network, ensure that an Ethernet connector or similar network adapter is attached to your system.**

2. **If you want to install a system that is connected through a `tip`(1) line, ensure that your window display is at least 80 columns wide and 24 rows long.**

   To determine the current dimensions of your `tip` window, use the `stty`(1) command.

3. **If you are using a profile diskette, insert the profile diskette into the system's diskette drive.**

   Note – The profile diskette contains a copy of the Solaris Device Configuration Assistant in addition to profile information. If you are using PXE network boot to boot the system over the network, you must configure your system so that the system boots from the network and not from the diskette.

4. **If you are using the system's DVD-ROM or CD-ROM drive to install the Solaris 9 software, insert the Solaris** ***x86 Platform Edition*** **DVD or the Solaris Software 1 of 2** ***x86 Platform Edition*** **CD in the drive.**

5. **Decide how to boot the system.**

   - If you boot from the Solaris DVD or the Solaris Installation CD, insert the disc. Your system's BIOS must support booting from a DVD or CD.
   - If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the systems's BIOS setup tool or the network adapter's configuration setup tool.
   - If you boot from a Diskette – Use the profile diskette that you inserted into the drive in Step 3 or insert the Solaris Device Configuration Assistant *x86 Platform Edition* diskette into the system's diskette drive.

---

x86 only – You can copy the Device Configuration Assistant software to a diskette from the Solaris *x86 Platform Edition* DVD or Solaris Software 2 of 2 *x86 Platform Edition* CD by using the procedure in Appendix I.

---

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

   The Device Configuration Assistant identifies the system's devices.

7. **On the Boot Solaris screen, select the device from which to boot the system. Select DVD, CD, Net, or Disk.**

8. **At the prompt, perform one of the following:**

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

            1 Solaris Interactive
            2 Custom JumpStart

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

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

   To select the custom JumpStart method, perform one of the following actions:

   Note – If you do not type 2 or type a boot command within 30 seconds, the Solaris `suninstall` program begins. You can stop the timer by typing any key on the command line.

   - Type 2 and press Enter.
   - To specify the location of the custom JumpStart configuration files, type the following command.

     `b install` [*url*:**`ask`**] [**`dhcp`**] [**`nowin`**]

     url Specifies the location of the custom JumpStart files. You can specify a URL for files that are located in the following places:

     - Local hard disk

       `file://`*jumpstart_dir_path/compressed_config_file*

     - NFS server

       `nfs://`*server_name:IP_address/jumpstart_dir_path/compressed_config _file*

     - HTTP server

       `http://`*server_name:IP_address/jumpstart_dir_path/compressed_config_file&proxy_info*

If you placed a sysidcfg file in the compressed configuration file, you must specify the IP address of the server that contains the file, as in the following example:

```
http://131.141.2.32/jumpstart/config.tar
```

If you saved the compressed configuration file on an HTTP server that is behind a firewall, you must use a proxy specifier during boot. You do not need to specify an IP address for the server that contains the file. You must specify an IP address for the proxy server, as in the following example:

```
http://www.shadow.com/jumpstart/
config.tar&proxy=131.141.6.151
```

ask
: Specifies that the installation program prompt you to type the location of the compressed configuration file after the system boots and connects to the network.

If you bypass the prompt by pressing Return, the installation program interactively configures the network parameters. The installation program then prompts you for the location of the compressed configuration file. If you bypass the prompt by pressing Return, the Solaris suninstall program begins.

dhcp
: Specifies to use a DHCP server to obtain network installation information that is needed to boot the system.

If you do not specify to use a DHCP server, the system uses the /etc/bootparams file or the name service bootparams database.

nowin
: Specifies not to begin the X program. You do not need to use the X program to perform a custom JumpStart installation, so you can reduce the installation time by using the nowin option.

9. **If you did not preconfigure system information in the sysidcfg file, when prompted, answer the questions about system configuration.**

10. **Follow the instructions on the screen to install the software.**

    When the JumpStart program finishes installing the Solaris software, the system reboots automatically.

    After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

    - /var/sadm/system/logs
    - /var/sadm/install/logs

CHAPTER **30**

Installing With Custom JumpStart (Examples)

This chapter provides an example of setting up and installing Solaris software on both SPARC based and x86 based systems by using a custom JumpStart installation.

Sample Site Setup

Figure 30–1 shows the site setup for this example.

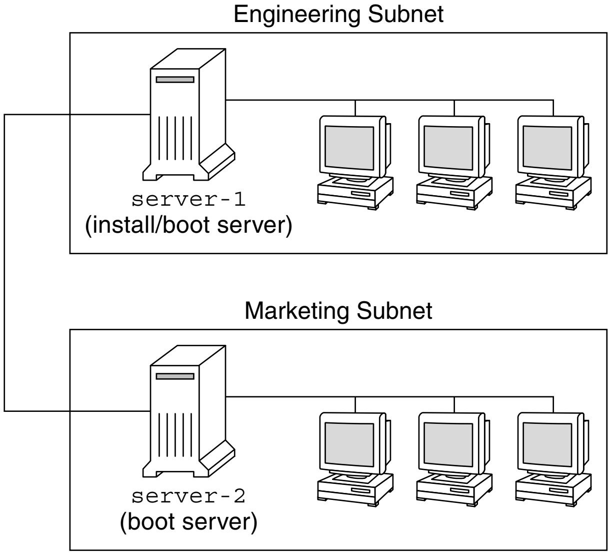


FIGURE 30–1 Sample Site Setup

At this sample site, the conditions are as follows:

- SPARC: The engineering group is located on its own subnet. This group uses SPARCstation™ systems for software development.
- x86: The marketing group is located on its own subnet. This group uses x86 based systems for running word processors, spreadsheets, and other office productivity tools.
- The site uses NIS. The Ethernet addresses, IP addresses, and host names of the systems are preconfigured in the NIS maps. The subnet mask, date and time, and geographic region for the site are also preconfigured in the NIS maps.

  Note – The peripheral devices for the marketing systems are preconfigured in the `sysidcfg` file.

- Both the engineering and marketing systems are to be installed with Solaris 9 software from the network.

Create an Install Server

Because the groups need to install Solaris 9 software from the network, you make `server-1` an install server for both groups. You use the `setup_install_server`(1M) command to copy the images to the `server-1` local disk (in the `/export/install` directory). Copy the images from the Solaris Software CDs and the Solaris Languages CD or from the Solaris DVD.

You must copy the image from the disc to an empty directory, in these examples the `sparc_9` and `x86_9` directories.

EXAMPLE 30–1 SPARC: Copying the Solaris 9 CDs

Insert the Solaris Software 1 of 2 *SPARC Platform Edition* CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# mkdir -p /export/install/sparc_9
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9
```

Insert the Solaris Software 2 of 2 *SPARC Platform Edition* CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

Insert the Solaris *SPARC Platform Edition* Languages CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

EXAMPLE 30–2 x86: Copying the Solaris 9 CDs

Insert the Solaris Software 1 of 2 *x86 Platform Edition* CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# mkdir -p /export/install/x86_9
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/x86_9
```

Insert the Solaris Software 2 of 2 *x86 Platform Edition* CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/x86_9
```

Insert the Solaris *x86 Platform Edition* Languages CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/x86_9
```

EXAMPLE 30–3 SPARC: Copying the Solaris 9 DVD

Insert the Solaris *SPARC Platform Edition* DVD in the DVD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# mkdir -p /export/install/sparc_9
server-1# cd /DVD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9
```

EXAMPLE 30–4 x86: Copying the Solaris 9 DVD

Insert the Solaris *x86 Platform Edition* DVD in the DVD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# mkdir -p /export/install/x86_9
server-1# cd /DVD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/x86_9
```

x86: Create a Boot Server for Marketing Systems

Systems cannot boot from an install server on a different subnet, so you make server-2 a boot server on the marketing group's subnet. You use the setup_install_server(1M) command to copy the boot software from the Solaris *x86 Platform Edition* DVD or the Solaris Software 1 of 2 *x86 Platform Edition* CD to the server-2 local disk (in the /export/boot directory).

If you insert the Solaris Software 1 of 2 *x86 Platform Edition* CD in the CD-ROM drive that is attached to server-2, type the following command:

```
server-2# cd /CD_mount_point/Solaris_9/Tools
server-2# ./setup_install_server -b /export/boot
```

If you insert the Solaris *x86 Platform Edition* DVD in the DVD-ROM drive that is attached to server-2, type the following command:

```
server-2# cd /DVD_mount_point/Solaris_9/Tools
server-2# ./setup_install_server -b /export/boot
```

In the setup_install_server command, -b specifies that setup_install_server is to copy the boot information to the directory that is named /export/boot.

Create a JumpStart Directory

Now that you have the install server and boot server set up, you create a JumpStart directory on `server-1`. You can use any system on the network. This directory holds files that are required for a custom JumpStart installation of Solaris software. You set up this directory by copying the sample directory from the Solaris DVD image or from the Solaris Software 1 of 2 CD image that has been copied to `/export/install`:

```
server-1# mkdir /jumpstart
server-1# cp -r /export/install/sparc_9/Solaris_9/Misc/jumpstart_sample  /jumpstart
```

Share the JumpStart Directory

To make the `rules` file and profiles accessible to systems on the network, you share the `/jumpstart` directory. To enable the sharing of a directory, you add the following line to the `/etc/dfs/dfstab` file:

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

Then, at the command line, you type the `shareall` command:

```
server-1# shareall
```

SPARC: Create the Engineering Group's Profile

For the engineering systems, you create a file that is named `eng_prof` in the `/jumpstart` directory. The `eng_prof` file contains the following entries, which define the Solaris 9 software to be installed on systems in the engineering group:

```
install_type  initial_install1
system_type   standalone2
partitioning  default3
cluster       SUNWCprog4
filesys       any 512 swap5
```

1. Specifies that the installation is to be treated as an initial installation, as opposed to an upgrade.
2. Specifies that the engineering systems are standalone systems.

3. Specifies that the JumpStart software uses default disk partitioning for installing Solaris software on the engineering systems.
4. Specifies that the Developer System Support software group is to be installed.
5. Specifies that each system in the engineering group is to have 512 Mbytes of swap space.

x86: Create the Marketing Group's Profile

For the marketing systems, you create a file that is named `marketing_prof` in the `/jumpstart` directory. The `marketing_prof` file contains the following entries, which define the Solaris 9 software to be installed on systems in the marketing group:

```
install_type  initial_install[1]
system_type   standalone[2]
partitioning  default[3]
cluster       SUNWCuser[4]
package       SUNWaudio[5]
```

1. Specifies that the installation is to be treated as an initial installation, as opposed to an upgrade.
2. Specifies that the marketing systems are standalone systems.
3. Specifies that the JumpStart software is to use default disk partitioning for installing Solaris on the marketing systems.
4. Specifies that the End User Solaris Software Group is to be installed.
5. Specifies that the audio demo software package is to be added to each system.

Update the `rules` File

Now you must add rules to the `rules` file. The Solaris `suninstall` program uses the rules to select the correct installation (profile) for each system during a custom JumpStart installation.

At this site, each department is located on its own *subnet* and has its own network address. The engineering department is located on subnet 255.222.43.0. The marketing department is located on 255.222.44.0. You can use this information to control how the engineering and marketing systems are installed with the Solaris 9 software. In the `/jumpstart` directory, you edit the `rules` file, delete all of the example rules, and add the following lines to the file:

```
network 255.222.43.0 - eng_prof -
network 255.222.44.0 - marketing_prof -
```

Basically, these rules state that systems on the 255.222.43.0 network are to be installed with the Solaris 9 software by using the eng_prof profile. The systems on the 255.222.44.0 network are to be installed with the Solaris 9 software by using the marketing_prof profile.

Note – You can use the sample rules to use a network address to identify the systems to be installed with the Solaris 9 software by using eng_prof and marketing_prof, respectively. You can also use host names, memory size, or model type as the rule keyword. Table 31–1 contains a complete list of keywords you can use in a rules file.

Validate the rules File

After the rules and profiles are set up, you run the check script to verify that the files are correct:

```
server-1# cd /jumpstart
server-1# ./check
```

If the check script does not find any errors, the script creates the rules.ok file.

SPARC: Set Up Engineering Systems to Install From the Network

After setting up the /jumpstart directory and files, you use the add_install_client command on the install server, server-1, to set up the engineering systems to install the Solaris 9 software from the install server. server-1 is also the boot server for the engineering group's subnet.

```
server-1# cd /export/install/sparc_9/Solaris_9/Tools
server-1# ./add_install_client -c server-1:/jumpstart host-eng1 sun4u
server-1# ./add_install_client -c server-1:/jumpstart host-eng2 sun4u
```

In the add_install_client command, the options that are used have the following meanings:

-c Specifies the server (server-1) and path (/jumpstart) to the JumpStart directory.

host-eng1 The name of a system in the engineering group.

host-eng2 The name of another system in the engineering group.

sun4u Specifies the platform group of the systems that use server-1 as an install server. The platform group is for Ultra 5 systems.

x86: Set Up Marketing Systems to Install From the Network

Next, you use the add_install_client command on the boot server (server-2). This command sets up the marketing systems to boot from the boot server and install the Solaris 9 software from the install server (server-1):

```
server-2# cd /marketing/boot-dir/Solaris_9/Tools
server-2# ./add_install_client -s server-1:/export/install/x86_9 \
-c server-1:/jumpstart host-mkt1 i86pc
server-2# ./add_install_client -s server-1:/export/install/x86_9 \
-c server-1:/jumpstart host-mkt2 i86pc
server-2# ./add_install_client -d -s server-1:/export/install/x86_9 \
-c server-1:/jumpstart SUNW.i86pc i86pc
```

In the add_install_client command, the options that are used have the following meanings:

-d Specifies that the client is to use DHCP to obtain the network install parameters. This option is required for clients to use PXE network boot to boot from the network. -d is optional for network boot clients that do not use PXE network boot.

-s Specifies the install server (server-1) and the path to the Solaris 9 software (/export/install/x86_9).

-c Specifies the server (server-1) and path (/jumpstart) to the JumpStart directory.

host-mkt1 The name of a system in the marketing group.

host-mkt2 The name of another system in the marketing group.

SUNW.i86pc The DHCP class name for all Solaris x86 clients. If you want to configure all Solaris x86 DHCP clients with a single command, use this class name.

i86pc Specifies the platform group of the systems that use this boot server. The platform name represents x86 based systems.

SPARC: Boot the Engineering Systems and Install Solaris 9 Software

After setting up the servers and files, you can boot the engineering systems by using the following `boot` command at the `ok` (PROM) prompt of each system:

```
ok boot net - install
```

The Solaris operating environment is automatically installed on the engineering group's systems.

x86: Boot the Marketing Systems and Install Solaris 9 Software

You can boot the system from one of the following:

- Solaris Software 1 of 2 *x86 Platform Edition* CD
- Solaris *x86 Platform Edition* DVD
- The network by using PXE network boot
- The profile diskette
- The Solaris Device Configuration Assistant *x86 Platform Edition* diskette

Solaris 9 is automatically installed on the marketing group's systems.

CHAPTER 31

Custom JumpStart (Reference)

This chapter lists keywords and values that you can use in the rules file, profiles, and begin and finish scripts.

- "Rule Keywords and Values" on page 335
- "Profile Keywords and Values" on page 340
- "Custom JumpStart Environment Variables" on page 369
- "Probe Keywords and Values" on page 372

Rule Keywords and Values

Table 31–1 describes the keywords and values that you can use in the rules file. For detailed instructions to create a rules file, see "Creating the rules File" on page 274.

TABLE 31–1 Descriptions of Rule Keywords and Values

Keyword	Value	Matches
any	minus sign (-)	Anything. The any keyword always succeeds.
arch	*processor_type*	A system's processor type.
	Valid values for *processor_type* are the following: ■ SPARC: sparc ■ x86: i386	The uname -p command reports the system's processor type.

TABLE 31–1 Descriptions of Rule Keywords and Values *(Continued)*

Keyword	Value	Matches

`disksize`	*actual_disk_name size_range*
	actual_disk_name – A disk name in the form `c`*x*`t`*y*`d`*z*, such as `c0t3d0`or `c0d0`, or the special word `rootdisk`. If `rootdisk` is used, the disk to be matched is determined in the following order:
	■ SPARC: The disk that contains the preinstalled boot image, which is a new SPARC based system with factory JumpStart installed
	■ The `c0t3d0s0` disk, if the disk exists
	■ The first available disk that is searched in kernel probe order
	size_range – The size of the disk, which must be specified as a range of Mbytes (*x*-*x*).
		The name and size of a system's disk in Mbytes.
		Example:
		`disksize c0t3d0 250-300`
		In the example, the JumpStart program attempts to match a system disk that is named `c0t3d0`. The disk can hold between 250 and 300 Mbytes of information.
		Example:
		`disksize rootdisk 750-1000`
		In the example, the JumpStart program first attempts to match a system disk that contains a preinstalled boot image. Next, the JumpStart program attempts to match the `c0t3d0s0` disk, if the disk exists. Finally, the JumpStart program attempts to match the first available disk that can hold between 750 Mbytes and 1 Gbyte of information.
		Note – When calculating *size_range*, remember that a Mbyte equals 1,048,576 bytes. A disk might be advertised as a "535–Mbyte" disk, but the disk might contain only 510 million bytes of disk space. The JumpStart program views the "535–Mbyte" disk as a 510–Mbyte disk because 535,000,000 / 1,048,576 = 510. A "535–Mbyte" disk does not match a *size_range* equal to 530–550.

`domainname`	*actual_domain_name*	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.
`hostaddress`	*actual_IP_address*	A system's IP address.

TABLE 31–1 Descriptions of Rule Keywords and Values *(Continued)*

Keyword	Value	Matches
`hostname`	*actual_host_name*	A system's host name. If you have a system that is 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, the JumpStart program attempts to match all of the system's disks in kernel probe order. If `rootdisk` is used, the disk to be matched is determined in the following order: ■ SPARC: The disk that contains the preinstalled boot image, which is a new SPARC based system with factory JumpStart installed ■ The `c0t3d0s0` disk, if the disk exists ■ The first available disk that is searched in kernel probe order *version* — A version name or the special words `any` or `upgrade`. If `any` is used, any Solaris or SunOS release is matched. If `upgrade` is used, any Solaris 2.6 or compatible release that can be upgraded is matched. If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is `SystemV`.	A disk that has a root (`/`) file system that corresponds to a particular version of Solaris software. Example: `installed c0t3d0s1 Solaris_9` In the example, the JumpStart program attempts to match a system that has a Solaris 9 root (`/`) file system on `c0t3d0s1`.
`karch`	*actual_platform_group* Valid values are `sun4m`, `sun4u`, `i86pc`, `prep`. A list of systems and their corresponding platform group is presented in the *Solaris 9 Sun Hardware Platform Guide*.	A system's platform group. If you have a system that is already installed, the `arch -k` command or the `uname -m` command reports the system's platform group.

TABLE 31–1 Descriptions of Rule Keywords and Values *(Continued)*

Keyword	Value	Matches
`memsize`	*physical_mem* The value must be a range of Mbytes, *x*-*x*, or a single Mbyte value.	A system's physical memory size in Mbytes. Example: `memsize 64-128` The example tries to match a system with a physical memory size between 64 and 128 Mbytes. If you have a system that is already installed, the output of the `prtconf` command, line 2, reports the system's physical memory size.
`model`	*actual_platform_name*	A system's platform name. See the *Solaris 9 Sun Hardware Platform Guide* for a list of valid platform names. To find the platform name of an installed system, use the `uname -i` command or the output of the `prtconf` command, line 5. **Note –** If the *actual_platform_name* contains spaces, you must replace spaces with underscores (_). Example: SUNW,Sun_4_50
`network`	*network_num*	A system's network number, which the JumpStart program determines by performing a logical AND between the system's IP address and the subnet mask. Example: `network 193.144.2.8` The example tries to match a system with a 193.144.2.8 IP address, if the subnet mask is 255.255.255.0.

TABLE 31–1 Descriptions of Rule Keywords and Values *(Continued)*

Keyword	Value	Matches
`osname`	`Solaris_`*x*	A version of Solaris software that is already installed on a system. Example: `osname Solaris_9` In the example, the JumpStart program attempts to match a system with the Solaris 9 operating environment already installed.
`probe`	*probe_keyword*	A valid probe keyword or a valid custom probe keyword. Example: `probe disks` The example returns the size of a system's disks in Mbytes and in kernel probe order, for example, `c0t3d0s1`, `c0t4d0s0`, on a SPARC system. The JumpStart program sets the `SI_DISKLIST`, `SI_DISKSIZES`, `SI_NUMDISKS`, and `SI_TOTALDISK` environment variables. **Note –** The `probe` keyword is unique in that the keyword does not attempt to match an attribute and run a profile. The `probe` keyword returns a value. Consequently, you cannot specify begin scripts, profiles, and finish scripts with the `probe` rule keyword. Probe keywords are described in Chapter 28.

TABLE 31–1 Descriptions of Rule Keywords and Values *(Continued)*

Keyword	Value	Matches
`totaldisk`	*size_range* The value must be specified as a range of Mbytes (*x*-*x*).	The total disk space on a system in Mbytes. The total disk space includes all the operational disks that are attached to a system. Example: `totaldisk 300-500` In the example, the JumpStart program tries to match a system with a total disk space between 300 and 500 Mbytes. **Note –** When calculating *size_range*, remember that one Mbyte equals 1,048,576 bytes. A disk might be advertised as a "535–Mbyte" disk, but the disk might have only 510 million bytes of disk space. The JumpStart program views the "535–Mbyte" disk as a 510–Mbyte disk because 535,000,000 / 1,048,576 = 510. A "535–Mbyte" disk does not match a *size_range* equal to 530–550.

Profile Keywords and Values

This section describes the profile keywords and values that you can use in a profile. For detailed instructions to create a profile, see "Creating a Profile" on page 277.

Profile Keywords at a Glance

Table 31–2 provides a quick way to determine which keywords you can use, based on your installation scenario. Unless otherwise noted in the keyword descriptions, the keyword can only be used with the initial installation option.

TABLE 31–2 Overview of Profile Keywords

	Installation Scenarios				
Profile Keywords	**Standalone System (Nonnetworked)**	**Standalone System (Networked) or Server**	**OS Server**	**Upgrade**	**Upgrade With Disk Space Reallocation**
`archive_location` (installing Solaris Flash archives)	✓	✓			
`backup_media`					✓
`boot_device`	✓	✓	✓		
`bootenv createbe`	✓	✓	✓		
`client_arch`			✓		
`client_root`			✓		
`client_swap`			✓		
`cluster` (adding software groups)	✓	✓	✓		
`cluster` (adding or deleting clusters)	✓	✓	✓	✓	✓
`dontuse`	✓	✓	✓		
`fdisk` (x86 only)	✓	✓	✓		
`filesys` (mounting remote file systems)		✓	✓		
`filesys` (creating local file systems)	✓	✓	✓		
`filesys` (creating mirrored file systems)	✓	✓	✓		
`forced_deployment` (installing Solaris Flash differential archives)	✓	✓			
`geo`	✓	✓	✓	✓	✓
`install_type`	✓	✓	✓	✓	✓
`isa_bits`	✓	✓	✓	✓	✓
`layout_constraint`					✓
`local_customization` (installing Solaris Flash archives)	✓	✓			

TABLE 31–2 Overview of Profile Keywords *(Continued)*

Profile Keywords	Installation Scenarios				
	Standalone System (Nonnetworked)	Standalone System (Networked) or Server	OS Server	Upgrade	Upgrade With Disk Space Reallocation
`locale`	✓	✓	✓	✓	✓
`metadb` (creating state database replicas)	✓	✓	✓		
`no_master_check` (installing Solaris Flash differential archives)	✓	✓			
`no_content_check` (installing Solaris Flash differential archives)	✓	✓			
`num_clients`			✓		
`package`	✓	✓	✓	✓	✓
`partitioning`	✓	✓	✓		
`root_device`	✓	✓	✓	✓	✓
`system_type`	✓	✓	✓		
`usedisk`	✓	✓	✓		

Profile Keyword Descriptions and Examples

`archive_location` Keyword

`archive_location` *retrieval_type location*

The values of *retrieval_type* and *location* depend on where the Solaris Flash archive is stored. The following sections contain the values you can use for *retrieval_type* and *location* and examples of how to use the `archive_location` keyword.

- "NFS Server" on page 343
- "HTTP or HTTPS Server" on page 343
- "FTP Server" on page 344
- "Local Tape" on page 345
- "Local Device" on page 346
- "Local File" on page 347

NFS Server

If the archive is stored on an NFS server, use the following syntax for the `archive_location` keyword.

`archive_location nfs` *server_name:/path/filename retry n*

server_name The name of the server where you stored the archive.

path The location of the archive to be retrieved from the specified server. If the path contains $HOST, the Solaris Flash installation utilities replace $HOST with the name of the clone system that you are installing.

filename The name of the Solaris Flash archive file.

retry n An optional keyword. *n* is the maximum number of times the Solaris Flash utilities attempt to mount the archive.

Examples:

```
archive_location nfs golden:/archives/usrarchive
```

```
archive_location nfs://golden/archives/usrarchive
```

HTTP or HTTPS Server

If the archive is stored on an HTTP server, use the following syntax for the `archive_location` keyword.

`archive_location http://`*server_name:port path/filename optional_keywords*

If the archive is stored on an HTTPS server, use the following syntax for the `archive_location` keyword.

`archive_location https://`*server_name:port path/filename optional_keywords*

server_name The name of the server where you stored the archive. *server_name* can be a port number or the name of a TCP service that has a port number that is determined at runtime.

port An optional port. If you do not specify a port, the Solaris Flash installation utilities use the default HTTP port number, 80.

path The location of the archive to be retrieved from the specified server. If the path contains $HOST, the Solaris Flash installation utilities replace $HOST with the name of the clone system that you are installing.

filename The name of the Solaris Flash archive file.

optional_keywords The optional keywords that you can specify when you retrieve a Solaris Flash archive from an HTTP server.

TABLE 31–3 Optional Keywords to Use With `archive_location` `http`

Keywords	Value Definitions
`auth basic` *user_name password*	If the archive is located on an HTTP server that is password protected, you must include the user name and password that you need to access the HTTP server in the profile file. **Note –** The use of this authentication method in a profile that is intended for use with custom JumpStart is risky. Unauthorized users might have access to the profile file that contains the password.
`timeout` *min*	The `timeout` keyword enables you to specify, in minutes, the maximum length of time that is allowed to pass without receipt of data from the HTTP server before the connection is closed, reopened, and resumed from the point where the timeout occurred. If you specify a `timeout` value of 0 (zero), the connection is not reopened because of inactivity. If a time-out reconnection occurs, the Solaris Flash installation utilities attempt to resume the installation at the last known position in the archive. If the Solaris Flash installation utilities cannot resume the installation at the last known position, the retrieval restarts from the beginning of the archive and the data that was retrieved prior to the timeout is discarded.
`proxy` *host:port*	The `proxy` keyword allows you to specify a proxy host and proxy port. You can use a proxy host to retrieve a Solaris Flash archive from the other side of a firewall. You must supply a proxy port when you specify the `proxy` keyword.

Example:

```
archive_location http://silver/archives/usrarchive.flar timeout 5
```

Example of the `auth basic` *user_name password* keyword:

```
archive_location http://silver/archives/usrarchive.flar timeout 5 user1 secret
```

FTP Server

If the archive is stored on an FTP server, use the following syntax for the `archive_location` keyword.

`archive_location ftp://`*user_name:password@server_name:port path/filename optional_keywords*

user_name:password The user name and password that you need to access the FTP server in the profile file.

server_name The name of the server where you stored the archive. *server_name* can be a port number or the name of a TCP service that has a port number that is determined at runtime.

port A is an optional port. If you do not specify a port, the Solaris Flash installation utilities use the default FTP port number, 21.

path The location of the archive to be retrieved from the specified server. If the path contains $HOST, the Solaris Flash installation utilities replace $HOST with the name of the clone system that you are installing.

filename The name of the Solaris Flash archive file.

optional_keywords The optional keywords that you can specify when you retrieve a Solaris Flash archive from an FTP server.

TABLE 31–4 Optional Keywords to Use With `archive_location` `ftp`

Keywords	Value Definitions
`timeout` *min*	The `timeout` keyword enables you to specify, in minutes, the maximum length of time that is allowed to pass without receipt of data from the FTP server before the connection is closed, reopened, and resumed from the point where the timeout occurred. If you specify a `timeout` value of 0 (zero), the connection is not reopened because of inactivity. If a time-out reconnection occurs, the Solaris Flash installation utilities attempt to resume the installation at the last known position in the archive. If the Solaris Flash installation utilities cannot resume the installation at the last known position, the retrieval restarts from the beginning of the archive and the data that was retrieved prior to the timeout is discarded.
`proxy` *host:port*	The `proxy` keyword allows you to specify a proxy host and proxy port. You can use a proxy host to retrieve a Solaris Flash archive from the other side of a firewall. You must supply a proxy port when you specify the `proxy` keyword.

Example:

```
archive_location ftp://user1:secret@silver/archives/usrarchive.flar timeout 5
```

Local Tape

If the archive is stored on a tape, use the following syntax for the `archive_location` keyword.

`archive_location local_tape` *device position*

device The name of the tape drive where you stored the Solaris Flash archive. If the device name is a canonical path, the Solaris Flash installation utilities retrieve the archive from the path to the device node. If you supply a device name that is not a canonical path, the Solaris Flash installation utilities add `/dev/rmt/` to the path.

position Designates the place on the tape drive where you saved the archive. If you do not supply a position, the Solaris Flash installation utilities retrieve the archive from the current position on the tape drive. By specifying a *position*, you can place a begin script or a `sysidcfg` file on the tape drive before the archive.

Examples:

```
archive_location local_tape /dev/rmt/0n 5
```

```
archive_location local_tape 0n 5
```

Local Device

You can retrieve a Solaris Flash archive from a local device if you stored the Solaris Flash archive on a file system-oriented, random-access device, such as a diskette or a CD-ROM. Use the following syntax for the `archive_location` keyword.

Note – You can retrieve an archive from stream-oriented devices, such as tape, by using the syntax for local tape.

`archive_location local_device` *device path/filename file_system_type*

device — The name of the drive where you stored the Solaris Flash archive. If the device name is a canonical path, the device is mounted directly. If you supply a device name that is not a canonical path, the Solaris Flash installation utilities add `/dev/dsk/` to the path.

path — The path to the Solaris Flash archive, relative to the root of the file system on the device you specified. If the path contains $HOST, the Solaris Flash installation utilities replace $HOST with the name of the clone system that you are installing.

filename — The name of the Solaris Flash archive file.

file_system_type — Specifies the type of file system on the device. If you do not supply a file system type, the Solaris Flash installation utilities attempt to mount a UFS file system. If the UFS mount fails, the Solaris Flash installation utilities attempt to mount an HSFS file system.

Examples:

To retrieve an archive from a local hard drive that is formatted as a UFS file system, use the following command:

```
archive_location local_device c0t0d0s0 /archives/$HOST
```

To retrieve an archive from a local CD-ROM that has an HSFS file system, use the following command:

```
archive_location local_device c0t0d0s0 /archives/usrarchive
```

Local File

You can retrieve an archive that you stored in the miniroot from which you booted the clone system as a local file. When you perform a custom JumpStart installation, you boot the system from a CD-ROM or an NFS-based miniroot. The installation software is loaded and run from this miniroot. Therefore, a Solaris Flash archive that you stored in the CD-ROM or NFS-based miniroot is accessible as a local file. Use the following syntax for the `archive_location` keyword.

`archive_location local_file` *path/filename*

path The location of the archive. The path must be accessible to the system as a local file while the system is booted from the Solaris Installation CD or from the Solaris DVD. The system cannot access `/net` when it is booted from the Solaris Installation CD or from the Solaris DVD.

filename The name of the Solaris Flash archive file.

Example:

```
archive_location local_file /archives/usrarchive
```

`backup_media` Profile Keyword

`backup_media` *type path*

Note – You can use `backup_media` only with the upgrade option when disk space reallocation is required.

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

Valid *type* Values	Valid *path* Values	Specification
`local_tape`	`/dev/rmt/`*n*	A local tape drive on the system that is being upgraded. *path* must be the character (raw) device path for the tape drive. *n* is the number of the tape drive.

Valid *type* Values	Valid *path* Values	Specification
`local_diskette`	`/dev/rdiskette`*n*	A local diskette drive on the system that is being upgraded. *path* must be the character (raw) device path for the diskette drive. *n* is the number of the diskette drive. Diskettes that you use for the backup must be formatted.
`local_filesystem`	`/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z* /*file_system*	A local file system on the system that is being upgraded. You cannot specify a local file system that is being changed by the upgrade. *path* can be a block device path for a disk slice. For example, the `t`*x* in `/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z* might not be needed. Or, *path* can be the absolute path to a file system that is mounted by the `/etc/vfstab` file.
`remote_filesystem`	*host*`:`/*file_system*	An NFS file system on a remote system. *path* must include the name or IP address of the remote system, *host*, and the absolute path to the NFS file system, *file_system*. The NFS file system must have read/write access.
`remote_system`	*user@host*`:`/*directory*	A directory on a remote system that can be reached by a remote shell, `rsh`. The system that is being upgraded must have access to the remote system through the remote system's `.rhosts` file. *path* must include the name of the remote system *host* and the absolute path to the directory *directory*. If a user login ID *user* is not specified, `root` is used by default.

Examples:

```
backup_media local_tape /dev/rmt/0
```

```
backup_media local_diskette /dev/rdiskette1
```

```
backup_media local_filesystem /dev/dsk/c0t3d0s4
```

```
backup_media local_filesystem /export
```

```
backup_media remote_filesystem system1:/export/temp
```

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

`boot_device` Profile Keyword

`boot_device` *device eeprom*

boot_device designates the device where the JumpStart program is to install the root (/) file system and the system's boot device.

If you do not specify the boot_device keyword in a profile, the following boot_device keyword is specified by default during the installation: boot_device any update.

device – Use one of the following values.

- SPARC: *cwtxdysz* or *cxdysz* – The disk slice where the JumpStart program places the root (/) file system, for example, c0t0d0s0.
- x86: *cwtxdy* or *cxdy* – The disk where the JumpStart program places the root (/) file system, for example, c0d0.
- existing – The JumpStart program places the root (/) file system on the system's existing boot device.
- any – The JumpStart program chooses where to place the root (/) file system. The JumpStart program attempts to use the system's existing boot device. The JumpStart program might choose a different boot device if necessary.

eeprom – Choose to update or preserve the system's EEPROM.

SPARC: Choose if you want to update or preserve the system's EEPROM to the specified boot device.

x86: You must specify the preserve value.

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

Note – SPARC: On SPARC systems, the *eeprom* value also allows you to update the system's EEPROM if you change the system's current boot device. By updating the system's EEPROM, the system can automatically boot from the new boot device.

Example:

```
boot_device c0t0d0s2 update
```

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

---

bootenv createbe Profile Keyword

`bootenv createbe bename` *new_BE_name* `filesystem` *mountpoint*`:`*device*`:`*fs_options* `[filesystem...]`

`bootenv createbe` enables you to create an empty-and-inactive boot environment at the same time you are installing the operating environment. At the least, you must create the root (/) file system. The slices are reserved for the file systems specified, but no file systems are copied. The boot environment is named, but not actually created until installed with a Solaris Flash archive. When the empty boot environment is installed with an archive, file systems are installed on the reserved slices. The following lists the values for *bename* and *filesystem*.

`bename` *new_BE_name*
: `bename` specifies the name of the new boot environment to be created. *new_BE_name* can be no longer than 30 characters, can contain only alphanumeric characters, and can contain no multibyte characters. The name must be unique on the system.

`filesystem` *mountpoint*:*device*:*fs_options*
: `filesystem` determines the type and number of file systems that are to be created in the new boot environment. At least one slice that contains the root (/) file system must be defined. File systems can be on the same disk or spread across multiple disks.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap slice.
- *device* must be available when the operating environment that is being installed is first booted. The device has no relation to JumpStart special storage devices such as *free*. The device cannot be a Solaris Volume Manager volume or Veritas Volume Manager volume. *device* is the name of a disk device, of the form `/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z*.
- *fs_options* can be one of the following:
  - `ufs`, which indicates a UFS file system.
  - `swap`, which indicates a swap file system. The swap mount point must be a – (hyphen).

For a profile example and background about using this keyword, see the following references:

- For an example of a profile, see Example 26–9.
- For background about using Solaris Live Upgrade that creates, upgrades, and activates inactive boot environments, see Chapter 32.
- For background about using a Solaris Flash archive, see Chapter 19.

client_arch Profile Keyword

`client_arch` *karch_value* `...`

client_arch specifies that the operating system server is to support a different platform group than the server uses. If you do not specify client_arch in the profile, any diskless client that uses the operating system server must contain the same platform group as the server. You must specify each platform group that you want the operating system server to support.

Valid values for *karch_value* are sun4m, sun4u, and i86pc. For a detailed list of platform names and various systems, see *Solaris 9 Sun Hardware Guide*.

Note – You can use client_arch only when system_type is specified as server.

---

client_root Profile Keyword

client_root *root_size*

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

Note – You can use client_root only when system_type is specified as server.

---

client_swap Profile Keyword

client_swap *swap_size*

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

Example:

```
client_swap 64
```

The example specifies that each diskless client is to have a swap space of 64 Mbytes.

Note – You can use client_swap only when system_type is specified as server.

How the Size of swap Is Determined

If a profile does not specify the size of swap, the JumpStart program determines the size of the swap space, based on the system's physical memory. Table 31–5 shows how the size of swap is determined during a custom JumpStart installation.

TABLE 31–5 Determining swap Size

Physical Memory (in Mbytes)	Swap Space (in Mbytes)
16–64	32
64–128	64
128–512	128
Greater than 512	256

The JumpStart program makes the size of swap no more than 20 percent of the disk where swap is located, unless the disk contains free space after laying out the other file systems. If free space exists, the JumpStart program allocates the free space to swap, and if possible, allocates the amount that is shown in Table 31–5.

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

cluster Profile Keyword (Adding Software Groups)

cluster *group_name*

cluster designates the software group to add to the system. The *group_name* for each software group is listed in the following table.

Software Group	*group_name*
Core System Support Software Group	SUNWCreq
End User Solaris Software Group	SUNWCuser
Developer Solaris Software Group	SUNWCprog
Entire Solaris Software Group	SUNWCall
Entire Solaris Software Group Plus OEM Support	SUNWCXall

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

cluster Profile Keyword (Adding or Deleting Clusters)

cluster *cluster_name add_delete_switch*

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

`cluster` designates whether a cluster is to be added or deleted from the software group that is to be installed on the system.

cluster_name must be in the form `SUNWC`*name*. To view detailed information about clusters and their names, start Admintool on an installed system and choose Software from the Browse menu.

add_delete_switch represents the option `add` or `delete`. Use *add_delete_switch* to indicate whether to add or delete the cluster that is specified. If you do not specify *add_delete_switch*, `add` is used by default.

When you use `cluster` (adding or deleting clusters) during an upgrade, the following conditions apply:

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

`dontuse` Profile Keyword

`dontuse` *disk_name* `...`

By default, the JumpStart program uses all of the operational disks on the system when `partitioning default` is specified. `dontuse` designates one or more disks that you do not want the JumpStart program to use. *disk_name* must be specified in the form `c`*x*`t`*y*`d`*z*or `c`*y*`d`*z*, for example, `c0t0d0`.

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

x86: `fdisk` Profile Keyword

`fdisk` *disk_name type size*

`fdisk` defines how the `fdisk` partitions are set up on an x86 based system. You can specify `fdisk` more than once. When `fdisk` partitions an x86 based system, the following occurs:

- All fdisk partitions on the disk are preserved unless you delete the partitions with the fdisk keyword by assigning *size* the value of delete or 0. Also, all existing fdisk partitions are deleted when *size* is set to all.
- A Solaris fdisk partition that contains a root (/) file system is always designated as the active partition on the disk.

---

x86 only – The system boots from the active partition by default.

- If the fdisk keyword is not specified in a profile, the following fdisk keyword is used by default during the installation.

```
fdisk all solaris maxfree
```

- fdisk entries are processed in the order in which the entries are listed in the profile.

disk_name – Use the following values to specify where the fdisk partition is to be created or deleted:

- c*x*t*y*d*z* or c*y*d*z* – A specific disk, for example, c0t3d0.
- rootdisk – The variable that contains the value of the system's root disk, which is determined by the JumpStart program as described in "How the System's Root Disk Is Determined" on page 368.
- all – All the selected disks.

type – Use the following values to specify the type of fdisk partition that is to 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 fdiskpartitions that are extended or reserved for data DOS. When you delete fdisk partitions by assigning *size* the value delete, dosprimary is an alias for the DOSHUGE, DOSOS12, and DOSOS16 fdisk types. When you create an fdisk partition, dosprimary is an alias for the DOSHUGE fdisk partition.
- *DDD* – An integer fdisk partition. *DDD* is an integer between 1 and 255 inclusive.

---

x86 only – You can specify this value only if *size* is delete.

- 0x*HH* – A hexadecimal fdisk partition. *HH* is a hexadecimal number between 01 and FF.

---

x86 only – You can specify this value only if *size* is delete.

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

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

size – Use one of the following values:

- *DDD* – An `fdisk` partition of size *DDD* in Mbytes is created on the specified disk. *DDD* must be an integer, and the JumpStart program automatically rounds the number up to the nearest cylinder boundary. Specifying a value of `0` is the same as specifying `delete`.
- `all` – An `fdisk` partition is created on the entire disk. All existing `fdisk` partitions are deleted.

  x86 only – The `all` 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.

  x86 only – The disk must contain at least one unused `fdisk` partition. Also, the disk must have free space or installation fails. The `maxfree` value can be specified only if *type* is `solaris` or `dosprimary`.

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

`filesys` Profile Keyword (Mounting Remote File Systems)

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

By using `filesys` with the listed values, the JumpStart program sets up the installed system to automatically mount remote file systems when the system boots. You can specify `filesys` more than once.

Example:

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

server: – The name of the server where the remote file system is located, 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 that is specified in *server*:*path*. If a name service is not running on the network, the *server_address* value can be used to populate the `/etc/hosts` file with the server's host name and IP address. If you are not specifying the server's IP address, you must specify a minus sign (`-`). For example, if you have a name service that is running on the network, you do not need to specify the server's IP address.

mount_pt_name – The name of the mount point on which the remote file system is to be mounted.

mount_options – One or more mount options, which is the same as the `-o` option of the `mount`(1M) command. The mount options 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 (`ro,quota` for example).

`filesys` Profile Keyword (Creating Local File Systems)

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

By using `filesys` with the values that are listed, the JumpStart program creates local file systems during the installation. You can specify `filesys` more than once.

slice – Use one of the following values:

- `any` – The JumpStart program places the file system on any disk.

Note – You cannot specify `any` 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 JumpStart program places the file system, for example, `c0t0d0s0` or `c0d0s0`.
- `rootdisk.s`*n* – The variable that contains the value for the system's root disk, which is determined by the JumpStart program as described in "How the System's Root Disk Is Determined" on page 368. The `s`*n* suffix indicates a specific slice on the disk.

size – Use one of the following values:

- *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 you use the `existing` value, you can change the name of an existing slice by specifying *file_system* as a different *mount_pt_name*.

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

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

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

file_system – The *file_system* value is optional and used when *slice* is specified as `any` or `c`*w*`t`*x*`d`*y*`s`*z*. If *file_system* is not specified, `unnamed` is set by default. If `unnamed` is set, you cannot specify the *optional_parameters* value. Use one of the following values:

- *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. The VTOC value is V_BACKUP. By default, slice 2 is an overlap slice that is a representation of the whole disk.

Note – You can specify `overlap` only when *size* is `existing`, `all`, or *start*:*size*.

- `unnamed` – The specified *slice* is defined as a raw slice, so *slice* does not have a mount–point name. If you do not specify *file_system*, `unnamed` is used by default.
- `ignore` – The specified *slice* is not used or recognized by the JumpStart program. You can use this option to specify that you want a file system to be ignored on a disk during installation. The JumpStart program creates a new file system on the same disk with the same name. You can use `ignore` only when `partitioning existing` is specified.

optional_parameters – Use one of the following values:

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

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

- *mount_options* – One or more mount options, which is the same as the `-o` option of the `mount`(1M) command. The mount options 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 space (`ro,quota`, for example).

`filesys` Profile Keyword (Creating Mirrored File Systems)

`filesys mirror[[:`*name*`]]`*slice* `[[`*slice*`]]` *size* *file_system* *optional_parameters*

By using the `filesys mirror` keywords with the values that are listed, the JumpStart program creates the RAID-1 and RAID-0 volumes that are necessary to create a mirrored file system. You can specify `filesys mirror` more than once to create RAID-1 volumes (mirrors) for different file systems.

Note – The `filesys mirror` keyword is only supported for initial installations.

name – This optional keyword enables you to name the RAID-1 volume (mirror.) Mirror names must start with the letter "d", followed by a number between 0 and 127, for example, `d100`. If you do not specify a mirror name, the custom JumpStart program assigns a mirror name for you. For guidelines about how to name mirrors, see "RAID Volume Name Requirements and Guidelines for Custom JumpStart and Solaris Live Upgrade" on page 116.

slice – This value specifies the disk slice where the custom JumpStart program places the file system you want to mirror. The slice value must follow the format c*w*t*x*d*y*s*z*, for example `c0t0d0s0` or `c0t0d0s5`. The custom JumpStart program creates a RAID-0 volume (single-slice concatentation) on the slice, and creates a RAID-1 volume to mirror the concatenation. You can specify up to two slices to mirror.

size – This value specifies the size, in MBytes, of the file system.

file_system – This value specifies the file system that you are mirroring. The custom JumpStart program creates the mirror from the slices that are specified and mounts the mirror on the specified file system. In addition to critical file systems, such as root (`/`), `/usr`, and `/var`, you can also specify `swap` as the file system.

optional_parameters – One or more mount options, which is the same as the `-o` option of the `mount`(1M) command. The mount options are added to the `/etc/vfstab` entry for the specified *file_system*. 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`.

For more information about creating mirrored file systems during your installation, see Chapter 10.

`forced_deployment` Profile Keyword (Installing Solaris Flash Differential Archives)

```
forced_deployment
```

`forced_deployment` forces the installation of a Solaris Flash differential archive onto a clone system that is different than the software expects.

Caution – If you use `forced_deployment`, all new files are deleted to bring the clone system to the expected state. If you are not certain that you want files deleted, use the default which protects new files by stopping the installation.

`geo` Profile Keyword

`geo` *locale*

geo designates the regional locale or locales that you want to install on a system or to add when upgrading a system. Values you can specify for *locale* are listed in the following table.

Value	Description
N_Africa	Northern Africa, including Egypt
C_America	Central America, including Costa Rica, El Salvador, Guatemala, Mexico, Nicaragua, Panama
N_America	North America, including Canada, United States
S_America	South America, including Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Paraguay, Peru, Uruguay, Venezuela
Asia	Asia, including Japan, Republic of Korea, People's Republic of China, Taiwan, Thailand
Ausi	Australasia, including Australia, New Zealand
C_Europe	Central Europe, including Austria, Czech Republic, Germany, Hungary, Poland, Slovakia, Switzerland
E_Europe	Eastern Europe, including Albania, Bosnia, Bulgaria, Croatia, Estonia, Latvia, Lithuania, Macedonia, Romania, Russia, Serbia, Slovenia, Turkey
N_Europe	Northern Europe, including Denmark, Finland, Iceland, Norway, Sweden
S_Europe	Southern Europe, including Greece, Italy, Portugal, Spain
W_Europe	Western Europe, including Belgium, France, Great Britain, Ireland, Netherlands
M_East	Middle East, including Israel

A complete list of the component locale values that compose each regional locale that is listed previously is presented in Appendix D.

Note – You can specify a geo keyword for each locale you need to add to a system.

install_type Profile Keyword

install_type *initial_upgrade_flash_switch*

install_type defines whether to erase and install a new Solaris operating environment on a system, upgrade the existing Solaris environment on a system, or install a Solaris Flash archive on the system.

Note – You must specify `install_type` in a profile, and `install_type` must be the first profile keyword in every profile.

You must use one of the following options for the *initial_upgrade_flash_switch*:

- `initial_install` – Specifies to perform an initial installation of the Solaris operating environment
- `upgrade` – Specifies to perform an upgrade of the Solaris operating environment
- `flash_install` – Specifies to install a Solaris Flash archive that overwrites all files
- `flash_update` – Specifies to install a Solaris Flash differential archive that overwrites only the files that are specified

Note – Some profile keywords can only be used with the `initial_install` option. Some profile keywords can only be used with the `upgrade` option. Some profile keywords can only be used with the `flash_install` option.

`isa_bits` Profile Keyword

`isa_bits` *bit_switch*

`isa_bits` specifies whether 64-bit or 32-bit Solaris 9 packages are to be installed.

bit_switch represents the option `64` or `32`, which you use to indicate whether 64-bit or 32-bit Solaris 9 packages are to be installed. If you do not set this keyword in the profile, the JumpStart program installs systems as follows:

- 64-bit packages on UltraSPARC™ systems
- 32-bit packages on all other systems

Note – If you use the `isa_bits` keyword, you must also use the latest `check` script in the `Solaris_9/Misc/jumpstart_sample` directory on the Solaris Software 1 of 2 CD or on the Solaris DVD.

`layout_constraint` Profile Keyword

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

Note – You can use `layout_constraint` only for the upgrade option when you need to reallocate disk space.

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

If you do not specify the `layout_constraint` keyword, the JumpStart program lays out the disk as follows:

- File systems that require more space for the upgrade are marked changeable.
- File systems that are on the same disk as the file system that requires more space and that are mounted by the `/etc/vfstab` file are marked changeable.
- Remaining file systems are marked fixed because auto-layout cannot change the file systems.

If you specify one or more `layout_constraint` keywords, the JumpStart program lays out the disk as follows:

- File systems that require more space for the upgrade are marked changeable.
- File systems for which you specified a `layout_constraint` keyword are marked with the specified constraint.
- The remaining file systems are marked fixed.

You cannot change the constraint on file systems that require more space for the upgrade because the file systems must be marked changeable. You can use the `layout_constraint` keyword to change the *minimum_size* values on file systems that require more space for the upgrade.

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

slice – *slice* specifies the file system's disk slice on which to specify the constraint. You must specify the system's disk slice in the form *cwtxdysz*or *cxdysz*.

constraint – Use one of the following constraints for the specified file system:

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

  When you mark a file system as changeable and *minimum_size* is not specified, the file system's minimum size is set to 10 percent more than the minimum size that is required. For example, if the minimum size for a file system is 100 Mbytes, the changed size is 110 Mbytes. If *minimum_size* is specified, any free space that remains, original size minus minimum size, is used for other file systems.

- movable – Auto-layout can move the file system to another slice on the same disk or different disk. The file system size remains the same.
- available – Auto-layout can use all of the space on the file system to reallocate space. All of the data in the file system is lost. The available constraint can only be specified on file systems that are not mounted by the /etc/vfstab file.
- collapse – Auto-layout moves and collapses the specified file system into the parent file system. You can use the collapse option to reduce the number of file systems on a system as part of the upgrade. For example, if a system has the /usr and /usr/share file systems, collapsing the /usr/share file system moves the file system into /usr, the parent file system. You can specify the collapse constraint only on file systems that are mounted by the /etc/vfstab file.

minimum_size – Specifies the size of the file system after auto-layout reallocates space. The *minimum_size* option enables you to change the size of a file system. The size of the file system might be larger if unallocated space is added to the file system. But, the size is never less than the value you specify. The *minimum_size* value is optional. Use this value only if you have marked a file system as changeable and the minimum size cannot be less than what the file system needs for the existing file system contents.

Examples:

```
layout_constraint c0t3d0s1 changeable 200

layout_constraint c0t3d0s4 movable

layout_constraint c0t3d1s3 available

layout_constraint c0t2d0s1 collapse
```

local_customization Profile Keyword (Installing Solaris Flash Archives)

local_customization *local directory*

Before you install a Solaris Flash archive on a clone system, you can create custom scripts to preserve local configurations on the clone system. The local_customization keyword designates the directory where you have stored these scripts. *local directory* is the path to the script on the clone system. For information on predeployment and postdeployment scripts, see "Creating Customization Scripts" on page 221.

locale Profile Keyword

locale *locale_name*

Note – You can use `locale` with both the initial installation and upgrade options.

`locale` designates the locale packages you want to install or add when upgrading for the specified *locale_name*. The *locale_name* values are the same as those values that are used for the `$LANG` environment variable. Appendix D contains a list of valid locale values.

When you use the `local` keyword, consider the following:

- If you have preconfigured a default locale, the locale is automatically installed. The English language packages are installed by default.
- You can specify a `locale` keyword for each locale you need to add to a system.

`metadb` Profile Keyword (Creating State Database Replicas)

`metadb` *slice* `[size` *size-in-blocks*`] [count` *number-of-replicas*`]`

The `metadb` keyword enables you to create Solaris Volume Manager state database replicas (metadbs) during your custom JumpStart installation. You can use the `metadb` keyword multiple times in your profile file to create state database replicas on different disk slices.

slice	You must specify the disk slice on which you want the custom JumpStart program to place the state database replica. The *slice* value must follow the format c*w*t*x*d*y*s*z*.
`size` *size-in-blocks*	The `size` optional keyword enables you to specify the size, in blocks, of the state database replica to be created. If you do not specify a `size` value, the custom JumpStart program uses a default size of 8192 blocks for the state database replica.
`count` *number-of-replicas*	You can specify the number of state database replicas you are creating by setting the optional `count` keyword value in your profile. If you do not specify a `count` value, the custom JumpStart program creates three state database replicas by default.

For more information about creating Solaris Volume Manager state database replicas during your installation, see “State Database Replicas Guidelines and Requirements” on page 114.

no_content_check Profile Keyword (Installing Solaris Flash Archives)

```
no_content_check
```

When installing a clone system with a Solaris Flash differential archive, you can use the no_content_check keyword to ignore file-by-file validation. File-by-file validation ensures that the clone system is a duplicate of the master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.

Caution – If you use no_content_check, all new files are deleted to bring the clone system to the expected state. If you are not certain that you want files deleted, use the default which protects new files by stopping the installation.

no_master_check Profile Keyword (Installing Solaris Flash Archives)

```
no_master_check
```

When installing a clone system with a Solaris Flash differential archive, you can use the no_master_check keyword to ignore checking the clone system to make sure it was built from the original master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system. For information on installing Solaris Flash differential archives, see "To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation" on page 317.

num_clients Profile Keyword

num_clients *client_num*

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

Note – You can use num_clients only when system_type is specified as server.

package Profile Keyword

package *package_name* [*add_delete_switch*]

Note – You can use package with both the initial installation and upgrade options.

package designates whether a package is to be added to or deleted from the software group that is to be installed on the system.

You must specify *package_name* in the form SUNW*name*. To view detailed information about packages and their names, on an installed system, use the pkginfo -l command.

add_delete_switch represents the option add or delete, which you use to indicate whether to add or delete the specified package. If you do not specify *add_delete_switch*, add is used by default.

Note – Some packages are required and cannot be deleted. You cannot individually add or delete localization packages by using the package profile keyword. To add localization packages, use the locale profile keyword.

When you use package for an upgrade, the JumpStart program performs the following actions:

- 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 not installed if the package is part of a cluster that is designated to be installed.

partitioning Profile Keyword

```
partitioning type
```

partitioning defines how the disks are divided into slices for file systems during the installation.

type – Use one of the following values:

- default – The JumpStart program selects the disks and creates the file systems on which to install the specified software, except for any file systems that are specified by the filesys keywords. rootdisk is selected first. The JumpStart program uses additional disks if the specified software does not fit on rootdisk.

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

  Note – When you use both the filesys and partitioning existing profile keywords, you must set size *size* to existing.

- explicit – The JumpStart program uses the disks and creates the file systems that are specified by the filesys keywords. If you specify only the root (/) file system with the filesys keyword, all of the Solaris software is installed in the root (/) file system.

  Note – If you use the explicit profile value, you must use the filesys keyword to specify the disks to use and file systems to create.

If you do not specify partitioning in the profile, the default type of partitioning is used by default.

root_device Profile Keyword

```
root_device slice
```

Note – You can use root_device with both the initial installation and upgrade options.

root_device designates the system's root disk. "How the System's Root Disk Is Determined" on page 368 contains additional information.

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

Example:

```
root_device c0t0d0s2
```

When you use the root_device keyword, consider the following:

- If you specify root_device on a system with only one disk, the root_device and the disk must match. Also, any filesys keywords that specify the root (/) file system must match root_device.
- If you are upgrading a RAID-1 volume (mirror), the value that is specified for root_device should be one side of the mirror. The other side of the mirror is automatically upgraded.

How the System's Root Disk Is Determined

A system's root disk is the disk on the system that contains the root (/) file system. In a profile, you can use the rootdisk variable in place of a disk name, which the JumpStart program sets to the system's root disk. Table 31–6 describes how the JumpStart program determines the system's root disk for the installation.

Note – The JumpStart program only determines a system's root disk size during an initial installation. You cannot change a system's root disk during an upgrade.

TABLE 31–6 How JumpStart Determines a System's Root Disk (Initial Installation)

Stage	Action
1	If the root_device keyword is specified in the profile, the JumpStart program sets rootdisk to the root device.
2	If rootdisk is not set and the boot_device keyword is specified in the profile, the JumpStart program sets rootdisk to the boot device.
3	If rootdisk is not set and a filesys *cwtxdysz size* / entry is specified in the profile, the JumpStart program sets rootdisk to the disk that is specified in the entry.
4	If rootdisk is not set and a rootdisk.s*n* entry is specified in the profile, the JumpStart program searches the system's disks in kernel probe order for an existing root file system on the specified slice. If a disk is found, the JumpStart program sets rootdisk to the found disk.
5	If rootdisk is not set and partitioning existing is specified in the profile, the JumpStart program searches the system's disks in kernel probe order for an existing root file system. If a root file system is not found or more than one is found, an error occurs. If a root file system is found, the JumpStart program sets rootdisk to the found disk.
6	If rootdisk is not set, the JumpStart program sets rootdisk to the disk where the root (/) file system is installed.

system_type Profile Keyword

system_type *type_switch*

system_type defines the type of system on which the Solaris environment is to be installed.

type_switch represents the option standalone or server, which you use to indicate the type of system on which the Solaris software is to be installed. If you do not specify system_type in a profile, standalone is used by default.

usedisk Profile Keyword

usedisk *disk_name* ...

By default, the JumpStart program uses all of the operational disks on the system when you specify partitioning default. The usedisk profile keyword designates one or more disks that you want the JumpStart program to use. You must specify *disk_name* in the form c*x*t*y*d*z* or c*y*d*z*, for example, c0t0d0 or c0d0s0.

If you specify usedisk in a profile, the JumpStart program uses only the disks that you specify after the usedisk keyword.

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

Custom JumpStart Environment Variables

You can use environment variables in your begin and finish scripts. For example, a begin script might extract the disk size, SI_DISKSIZES, and install or not install particular packages on a system, based on the actual disk size the script extracts.

Information that is gathered about a system is stored in these environment variables, which are generally set or not, depending on the rule keywords and values you use in the rules file.

For example, information about which operating system is already installed on a system is only available in SI_INSTALLED after the installed keyword is used.

Table 31–7 describes these variables and their values.

TABLE 31–7 Installation Environment Variables

Environment Variable	Value
CHECK_INPUT	The path to the rules file in the JumpStart directory, which is mounted on /tmp/install_config/rules.
HOME	The root's home directory during installation, which is /tmp/root.
PATH	The shell search path during installation, which is /sbin:/usr/sbin/install.d:/usr:/usr/sbin:/usr/bin.
SI_ARCH	The hardware architecture of the install client. The SI_ARCH variable is set when the arch keyword is used in the rules file.
SI_BEGIN	The name of the begin script, if one is used.
SI_CLASS	The name of the profile that is used to install the install client.
SI_CONFIG_DIR	The path to the JumpStart directory, which is mounted on /tmp/install_config. **Note –** The SI_CONFIG_DIR variable is not available for custom JumpStart installations that use the WAN boot installation method.
SI_CONFIG_FILE	The path to the rules file in the JumpStart directory, which is mounted on /tmp/install_config/rules.
SI_CONFIG_PROG	The rules file.
SI_CUSTOM_PROBES_FILE	The custom_probes.ok file, in which you can define your own rule and probe keywords. If you create a custom_probes.ok file, you can use the file to extend the list of default rule keywords, which are described in "Rule Keywords and Values" on page 335. You can also use the file to extend the list of default probe keywords that are described in "Probe Keywords and Values" on page 372.
SI_DISKLIST	A comma-separated list of disk names on the install client. The SI_DISKLIST variable is set when the disksize keyword is used and matched in the rules file. The SI_DISKLIST and SI_NUMDISKS variables are used to determine the physical disk to use for the rootdisk. rootdisk is described in "How the System's Root Disk Is Determined" on page 368.
SI_DISKSIZES	A comma-separated list of disk sizes on the install client. The SI_DISKSIZES variable is set when the disksize keyword is used and matched in the rules file.
SI_DOMAINNAME	The domain name. The SI_DOMAINNAME variable is set when the dommainname keyword is used and matched in the rules file.
SI_FINISH	The name of the finish script, if one is used.
SI_HOSTADDRESS	The install client's IP address.
SI_HOSTID	The install client's Ethernet address.

TABLE 31–7 Installation Environment Variables *(Continued)*

Environment Variable	Value
SI_HOSTNAME	The install client's host name. The SI_HOSTNAME variable is set when the hostname keyword is used and matched in the rules file.
SI_INSTALLED	The device name of a disk with a specific operating system on the disk, for example, Solaris, SunOS, or System V. The SI_INSTALLED variable is set when the installed keyword is used and matched in the rules file. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.
SI_INST_OS	The name of the operating system. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.
SI_INST_VER	The version of the operating system. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.
SI_KARCH	The install client's kernel architecture. The SI_KARCH variable is set when the karch keyword is used and matched in the rules file.
SI_MEMSIZE	The amount of physical memory on the install client. The SI_MEMSIZE variable is set when the memsize keyword is used and matched in the rules file.
SI_MODEL	The install client's model name. The SI_MODEL variable is set when the model keyword is used and matched in the rules file.
SI_NETWORK	The install client's network number. The SI_NETWORK variable is set when the network keyword is used and matched in the rules file.
SI_NUMDISKS	The number of disks on an install client. The SI_NUMDISKS variable is set when the disksize keyword is used and matched in the rules file. The SI_NUMDISKS and SI_DISKLIST variables are used to determine the physical disk to use for the rootdisk. rootdisk is described in "How the System's Root Disk Is Determined" on page 368.
SI_OSNAME	The operating system release on the Solaris 9 software image. For example, you can use the SI_OSNAME variable in a script if you are installing the Solaris software on systems that are based on the version of the operating system on the Solaris DVD or the Solaris Software 1 of 2 CD image.
SI_PROFILE	The path to the profile in the mounted JumpStart directory. The path is /tmp/install_config/*profile_name*. If you are creating a derived profile, SI_PROFILE is set to the /tmp/install.input file.
SI_ROOTDISK	The device name of the disk that is represented by the logical name rootdisk. The SI_ROOTDISK variable is set when the disksize or the installed keyword is set to rootdisk in the rules file.
SI_ROOTDISKSIZE	The size of the disk that is represented by the logical name rootdisk. The SI_ROOTDISKSIZE variable is set when the disksize or the installed keyword is set to rootdisk in the rules file.

TABLE 31–7 Installation Environment Variables *(Continued)*

Environment Variable	Value
SI_SYS_STATE	The /a/etc/.sysIDtool.state file. You can edit this file in a finish script to prevent the sysidroot program from prompting for a root password before the system reboots.
SI_TOTALDISK	The total amount of disk space on the install client. The SI_TOTALDISK variable is set when the totaldisk keyword is used and matched in the rules file.
SHELL	The default shell during installation, which is /sbin/sh.
TERM	The install client's terminal type.
TZ	The default time zone, as specified in the NIS or NIS+ name service.

Probe Keywords and Values

Table 31–8 describes each rule keyword and its equivalent probe keyword.

Note – Always place probe keywords at or near the beginning of the rules file.

TABLE 31–8 Descriptions of Probe Keywords

Rule Keyword	Equivalent Probe Keyword	Description of Probe Keyword
any	None	
arch	arch	Determines the kernel architecture,i386 or SPARC, and sets SI_ARCH.
disksize	disks	Returns the size of a system's disks in Mbytes in kernel probe order, c0t3d0s0, c0t3d0s1, c0t4d0s0. disksize sets SI_DISKLIST, SI_DISKSIZES, SI_NUMDISKS, and SI_TOTALDISK.
domainname	domainname	Returns a system's NIS or NIS+ domain name or blank and sets SI_DOMAINNAME. The domainname keyword returns the output of domainname(1M).
hostaddress	hostaddress	Returns a system's IP address, the first address that is listed in the output of ifconfig(1M) -a that is not lo0, and sets SI_HOSTADDRESS.
hostname	hostname	Returns a system's host name that is the output from uname(1) -n and sets SI_HOSTNAME.

TABLE 31–8 Descriptions of Probe Keywords *(Continued)*

Rule Keyword	Equivalent Probe Keyword	Description of Probe Keyword
`installed`	`installed`	Returns the version name of the Solaris operating environment that is installed on a system and sets `SI_ROOTDISK` and `SI_INSTALLED`. If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is `SystemV`.
`karch`	`karch`	Returns a system's platform group, for example i86pc, sun4m, and sun4, and sets `SI_KARCH`. For a list of platform names, see *Solaris 9 Sun Hardware Platform Guide*.
`memsize`	`memsize`	Returns the size of physical memory on a system in Mbytes and sets `SI_MEMSIZE`.
`model`	`model`	Returns a system's platform name and sets `SI_MODEL`. For a list of platform names, see the *Solaris 9 Sun Hardware Platform Guide*.
`network`	`network`	Returns a system's network number, which the JumpStart program determines by performing a logical AND between the system's IP address and the subnet mask. The system's IP address and the subnet mask are extracted from the first address that is listed in the output of `ifconfig`(1M) `-a` that is not lo0. The `network` keyword sets `SI_NETWORK`.
`osname`	`osname`	Returns the version and operating system name of the Solaris operating environment that is found on a CD and sets `SI_OSNAME`. If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is `SystemV`.
	`rootdisk`	Returns the name and size in Mbytes of a system's root disk and sets `SI_ROOTDISK`.
`totaldisk`	`totaldisk`	Returns the total disk space on a system (in Mbytes) and sets `SI_TOTALDISK`. The total disk space includes all of the operational disks that are attached to a system.

CHAPTER **32**

Solaris Live Upgrade (Topics)

This section provides instructions for using Solaris Live Upgrade to create and upgrade an inactive boot environment. The boot environment can then be switched to become the active boot environment.

Chapter 33	Provides overview information on the Solaris Live Upgrade process.
Chapter 34	Provides information that you need to know before creating a boot environment.
Chapter 35	Provides step-by-step instructions for installing Solaris Live Upgrade, using the menus, and creating a boot environment.
Chapter 36	Provides step-by-step instructions for an operating system upgrade or installing a Solaris Flash archive on a boot environment, switching a boot environment to make it active, and recovering quickly from a failed upgrade.
Chapter 37	Provides step-by-step instructions for maintaining a boot environment and viewing status.
Chapter 38	Provides examples of a Solaris Live Upgrade.
Chapter 39	Lists Solaris Live Upgrade commands.

CHAPTER 33

Solaris Live Upgrade (Overview)

This chapter describes the Solaris Live Upgrade process.

Note – This book uses the term *slice*, but some Solaris documentation and programs might refer to a slice as a partition.

Solaris Live Upgrade Introduction

Solaris Live Upgrade provides a method of upgrading a system while the system continues to operate. While your current boot environment is running, you can duplicate the boot environment, then upgrade the duplicate. Or, rather than upgrading, you can install a Solaris Flash archive on a boot environment. The original system configuration remains fully functional and unaffected by the upgrade or installation of an archive. When you are ready, you can activate the new boot environment by rebooting the system. If a failure occurs, you can quickly revert to the original boot environment with a simple reboot. This switch eliminates the normal downtime of the test and evaluation process.

Solaris Live Upgrade enables you to duplicate a boot environment without affecting the currently running system. You can then do the following:

- Upgrade a system.
- Change the current boot environment's disk configuration to different file system types, sizes, and layouts on the new boot environment.
- Maintain numerous boot environments with different images. For example, you can create one boot environment that contains current patches and create another boot environment that contains an Update release.

Some understanding of basic system administration is necessary before using Solaris Live Upgrade. For background information on system administration tasks such as managing file systems, mounting, booting, and managing swap, see the *System Administration Guide: Devices and File Systems*.

Solaris Live Upgrade Process

The following is an overview of the tasks necessary to create a copy of the current boot environment, upgrade the copy, and switch the upgraded copy to become the active boot environment.

Creating a Boot Environment

The process of creating a boot environment provides a method of copying critical file systems from an active boot environment to a new boot environment. The disk is reorganized if necessary, file systems are customized, and the critical file systems are copied to the new boot environment.

File System Types

Solaris Live Upgrade distinguishes between two file system types: critical file systems and shareable file systems. Critical file systems are required by the Solaris operating environment. These file systems are separate mount points in the `vfstab` of the active and inactive boot environments. Examples are root (`/`), `/usr`, `/var`, or `/opt`. These file systems are always copied from the source to the inactive boot environment. Critical file systems are sometimes referred to as *non-shareable*. Shareable file systems are user-defined files such as `/export` that contain the same mount point in the `vfstab` in both the active and inactive boot environments. Therefore, updating shared files in the active boot environment also updates data in the inactive boot environment. When you create a new boot environment, shareable file systems are shared by default. But you can specify a destination slice and then the file systems are copied. For more detailed information about shareable file systems, see "Guidelines for Selecting Slices for Shareable File Systems" on page 404.

Swap is a special shareable file system. Like a shareable file system, all swap slices are shared by default. But, if you specify a destination directory for swap, the swap slice is copied. For procedures on reconfiguring swap, see the following:

- "To Create a Boot Environment (Character Interface)" Step 9
- "To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)" on page 426

Solaris Live Upgrade can create a boot environment with RAID-1 volumes (mirrors) on file systems. For an overview, see “Creating a Boot Environment With Mirrored File Systems” on page 382.

Copying File Systems

The process of creating a new boot environment begins by identifying an unused slice where a critical file system can be copied. If a slice is not available or a slice does not meet the minimum requirements, you need to format a new slice.

After the slice is defined, you can reconfigure the file systems on the new boot environment before the file systems are copied into the directories. You reconfigure file systems by splitting and merging them, which provides a simple way of editing the `vfstab` to connect and disconnect file system directories. You can merge file systems into their parent directories by specifying the same mount point. You can also split file systems from their parent directories by specifying different mount points.

After file systems are configured on the inactive boot environment, you begin the automatic copy. Critical file systems are copied to the designated directories. Shareable file systems are not copied, but are shared. The exception is that you can designate some shareable file systems to be copied. When the file systems are copied from the active to the inactive boot environment, the files are directed to the new directories. The active boot environment is not changed in any way.

- For procedures about splitting and merging file systems, see the following procedures:
  - “To Create a Boot Environment (Character Interface)” Step 7 or Step 8
  - “Creating a New Boot Environment” on page 414
  - “To Create a Boot Environment and Split File Systems (Command-Line Interface)” on page 424
- For an overview of creating a boot environment with mirrored file systems, see “Creating a Boot Environment With Mirrored File Systems” on page 382.

The following figures illustrate various ways of creating new boot environments.

Figure 33–1 shows that critical file system root (`/`) has been copied to another slice on a disk to create a new boot environment. The active boot environment contains root (`/`) on one slice. The new boot environment is an exact duplicate with root (`/`) on a new slice. The file systems `/swap` and `/export/home` are shared by the active and inactive boot environments.

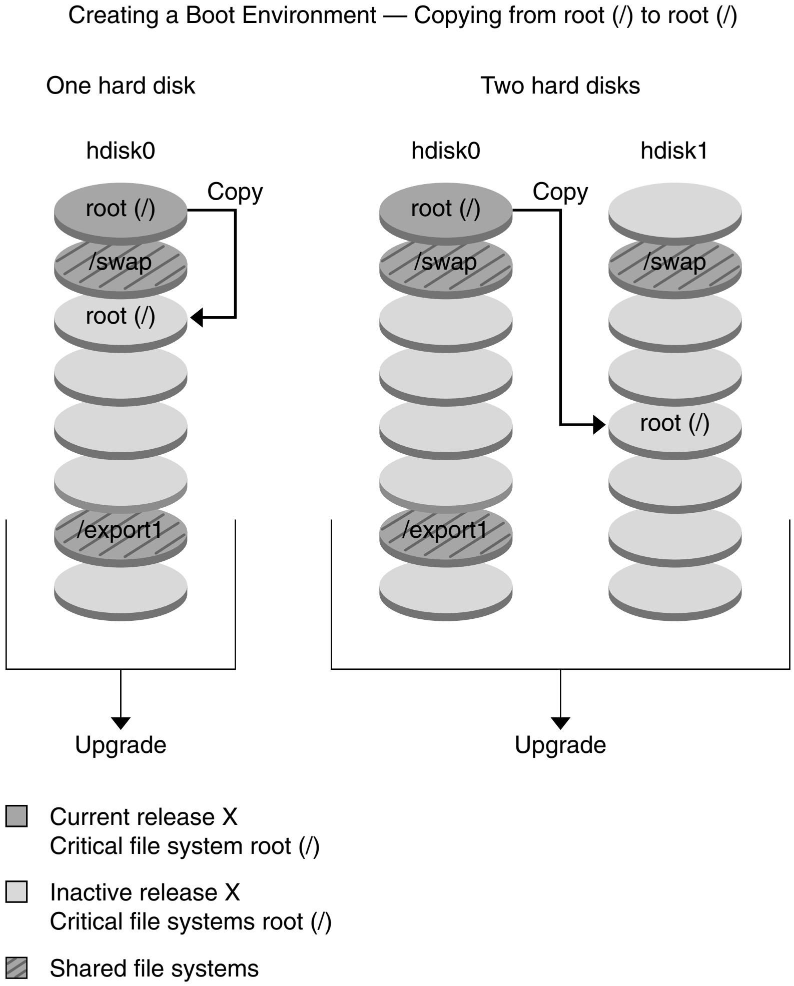


FIGURE 33–1 Creating an Inactive Boot Environment - Copying root (/) to root (/)

Figure 33–2 shows critical file systems that have been split and have been copied to slices on a disk to create a new boot environment. The active boot environment contains root (/) on one slice. On that slice, root (/) contains the `/usr`, `/var`, and `/opt` directories. In the new boot environment, root (/) is split and `/usr` and `/opt` are put on separate slices. The file systems `/swap` and `/export/home` are shared by both boot environments.

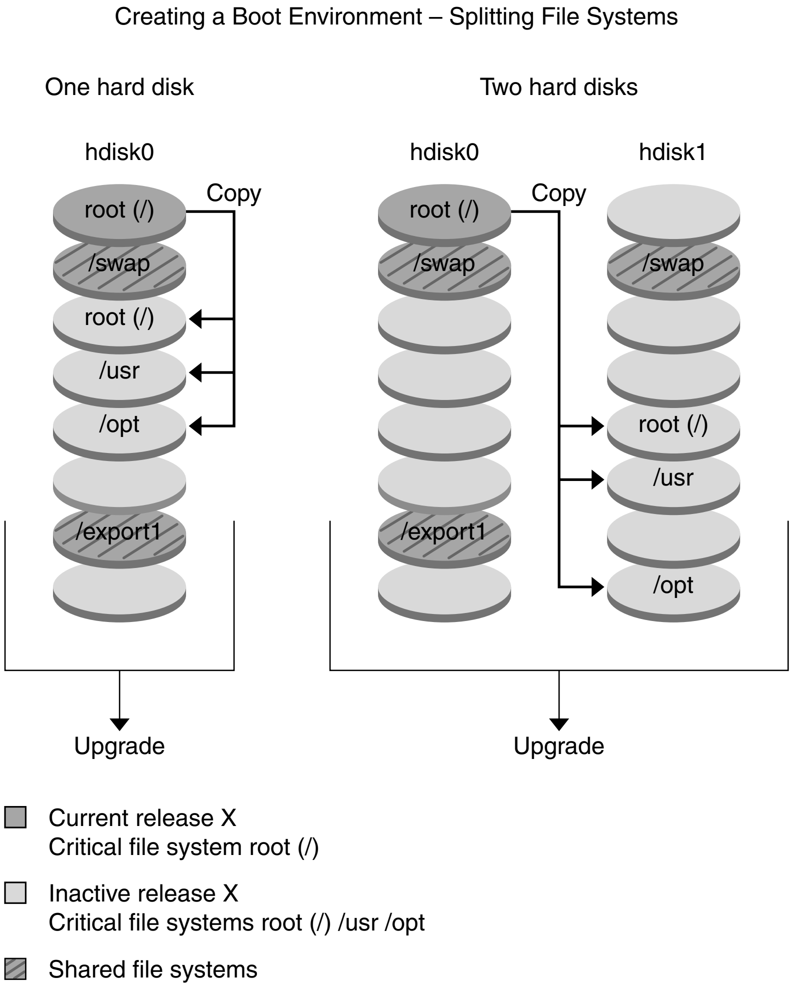


FIGURE 33–2 Creating an Inactive Boot Environment - Splitting File Systems

Figure 33–3 shows critical file systems that have been merged and have been copied to slices on a disk to create a new boot environment. The active boot environment contains root (`/`), `/usr`, `/var`, and `/opt` with each file system on their own slice. In the new boot environment, /usr and `/opt` are merged into root (`/`) on one slice. The file systems `/swap` and `/export/home` are shared by both boot environments.

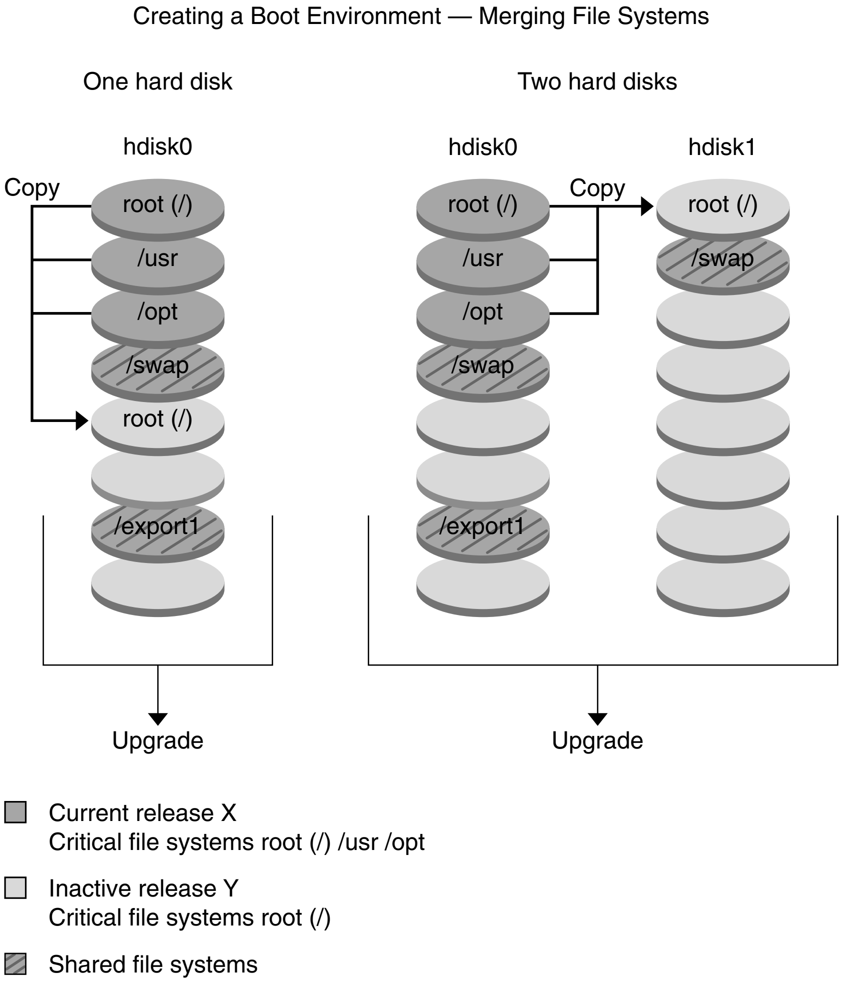


FIGURE 33–3 Creating an Inactive Boot Environment - Merging File Systems

Creating a Boot Environment With Mirrored File Systems

Solaris Live Upgrade uses Solaris Volume Manager technology to create a boot environment that can contain mirrored file systems. Solaris Volume Manager provides a powerful way to reliably manage your disks and data by using volumes. Solaris Volume Manager enables concatenations, stripes, and other complex configurations. Solaris Live Upgrade enables a subset of these tasks, such as creating a RAID-1 volume for the root (/) file system.

A volume can group disk slices across several disks to transparently appear as a single disk to the operating environment. Solaris Live Upgrade is limited to creating a boot environment for the root (/) file system that contains single-slice concatenations inside a RAID-1 volume (mirror). This limitation is because the boot PROM is restricted to choosing one slice from which to boot.

When creating a boot environment, you can use Solaris Live Upgrade to manage the following tasks.

- Detach a single-slice concatenation (submirror) from a RAID-1 volume (mirror). The contents can be preserved to become the content of the new boot environment if necessary. Because the contents are not copied, the new boot environment can be quickly created. After the submirror is detached from the original mirror, the submirror is no longer part of the mirror. Reads and writes on the submirror are no longer performed through the mirror.
- Create a boot environment that contains a mirror.
- Attach a maximum of three single-slice concatenations to the newly created mirror.

To use the mirroring capabilities of Solaris Live Upgrade, you must create a least one state database and at least three state database replicas. A state database stores information on disk about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a state database replica. When you copy the state database, you protect against data loss from single points of failure. For procedures about creating a state database, see "State Database (Overview)" in *Solaris Volume Manager Administration Guide*.

You use the `lucreate` command with the `-m` option to create a mirror, detach submirrors, and attach submirrors for the new boot environment.

- For procedures, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.
- For an overview of mirroring when installing, see Chapter 10.
- For in-depth information about other complex Solaris Volume Manager configurations that are not supported if using Solaris Live Upgrade, see "Storage Management Concepts" in *Solaris Volume Manager Administration Guide*.

TABLE 33–1 Solaris Volume Manager Elements Used by Solaris Live Upgrade

Term	Description
state database	A state database stores information on disk about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a state database replica. The state database tracks the location and status of all known state database replicas.

TABLE 33–1 Solaris Volume Manager Elements Used by Solaris Live Upgrade *(Continued)*

Term	Description
state database replica	A copy of a state database. The replica ensures that the data in the database is valid.
volume	A group of physical slices or other volumes that appear to the system as a single logical device. A volume is functionally identical to a physical disk in the view of an application or file system. In some command-line utilities, a volume is called a metadevice.

Table 33–2 shows the components that Solaris Live Upgrade can manage.

TABLE 33–2 Classes of Volumes

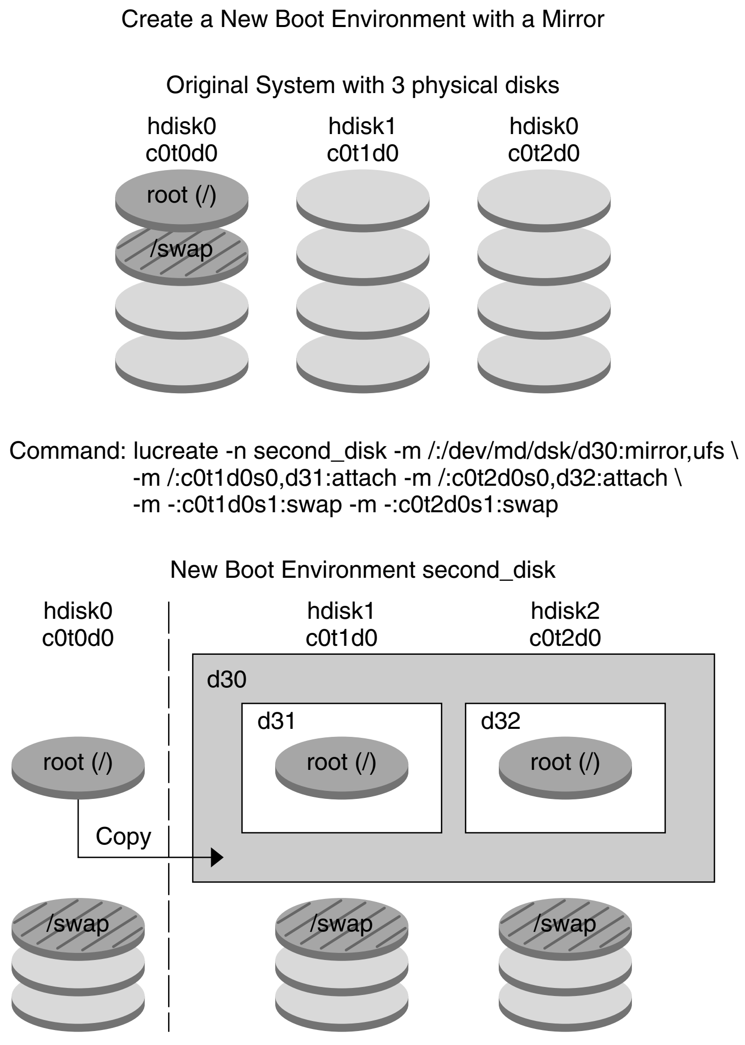
Term	Description
RAID-1 volume	A class of volume that replicates data by maintaining multiple copies. A RAID-1 volume is sometimes called a mirror. A RAID-1 volume is composed of one or more RAID-0 volumes that are called submirrors.
RAID-0 volume	A class of volume that can be a stripe or a concatenation. These components are also called submirrors. A stripe or concatenation is the basic building blocks for mirrors.
mirror	A RAID-1 volume. See RAID-1 volume.
concatenation	A RAID-0 volume. If slices are concatenated, the data is written to the first available slice until that slice is full. When that slice is full, the data is written to the next slice, serially. A concatenation provides no data redundancy unless it is contained in a mirror.
submirror	See RAID-0 volume.

Figure 33–4 shows a new boot environment with a RAID-1 volume (mirror) that is created on two physical disks. The following command created the new boot environment and the mirror.

```
# lucreate -n second_disk -m /:/dev/md/dsk/d30:mirror,ufs \
-m /:c0t1d0s0,d31:attach -m /:c0t2d0s0,d32:attach \
-m -:c0t1d0s1:swap -m -:c0t2d0s1:swap
```

This command performs the following tasks:

- Creates a new boot environment, `second_disk`.
- Creates a mirror `d30` and configures a UFS file system.
- Creates a single-device concatenation on slice 0 of each physical disk. The concatenations are named `d31` and `d32`.
- Adds the two concatenations to mirror `d30`.
- Copies the root (`/`) file system to the mirror.
- Configures files systems for swap on slice 1 of each physical disk.



```
Command: lucreate -n second_disk -m /:/dev/md/dsk/d30:mirror,ufs \
         -m /:c0t1d0s0,d31:attach -m /:c0t2d0s0,d32:attach \
         -m -:c0t1d0s1:swap -m -:c0t2d0s1:swap
```

d30 — RAID-1 volume (mirror)

d31 — Single-slice concatenation (submirror)

d32 — Single-slice concatenation (submirror)

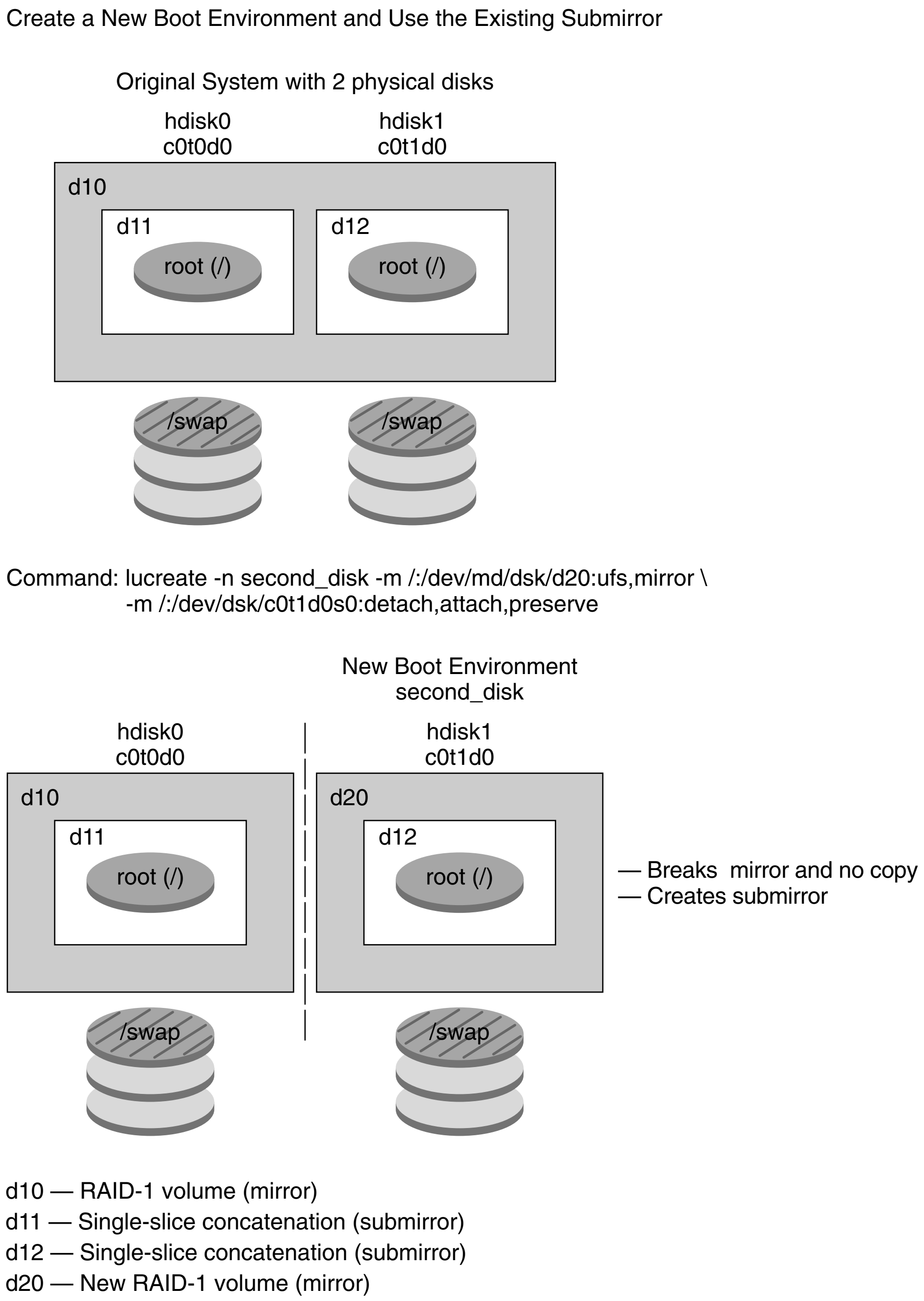
FIGURE 33–4 Create a Boot Environment and Create a Mirror

Figure 33–5 shows a new boot environment that contains a RAID-1 volume (mirror). The following command created the new boot environment and the mirror.

```
# lucreate -n second_disk -m /:/dev/md/dsk/d20:ufs,mirror \
-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

This command performs the following tasks:

- Creates a new boot environment, `second_disk`.
- Breaks mirror `d10` and detaches concatenation `d12`.
- Preserves the contents of concatenation `d12` and file systems are not copied.
- Creates a new mirror `d20`. You now have two one-way mirrors `d10` and `d20`.
- Attaches concatenation d12 to mirror d20.



```
Command: lucreate -n second_disk -m /:/dev/md/dsk/d20:ufs,mirror \
         -m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

d10 — RAID-1 volume (mirror)
d11 — Single-slice concatenation (submirror)
d12 — Single-slice concatenation (submirror)
d20 — New RAID-1 volume (mirror)

FIGURE 33–5 Create a Boot Environment and Use the Existing Submirror

Upgrading a Boot Environment

After you have created a boot environment, you can perform an upgrade on the boot environment. As part of that upgrade, the boot environment can contain RAID-1 volumes (mirrors) for any file systems. The upgrade does not affect any files in the active boot environment. When you are ready, you activate the new boot environment, which then becomes the current boot environment.

- For procedures about upgrading a boot environment, see Chapter 36.
- For an example of upgrading a boot environment with mirrored file systems, see "Example of Detaching and Upgrading One Side of a RAID 1 Volume (Mirror) (Command-Line Interface)" on page 490.

Figure 33–6 shows an upgrade to an inactive boot environment.

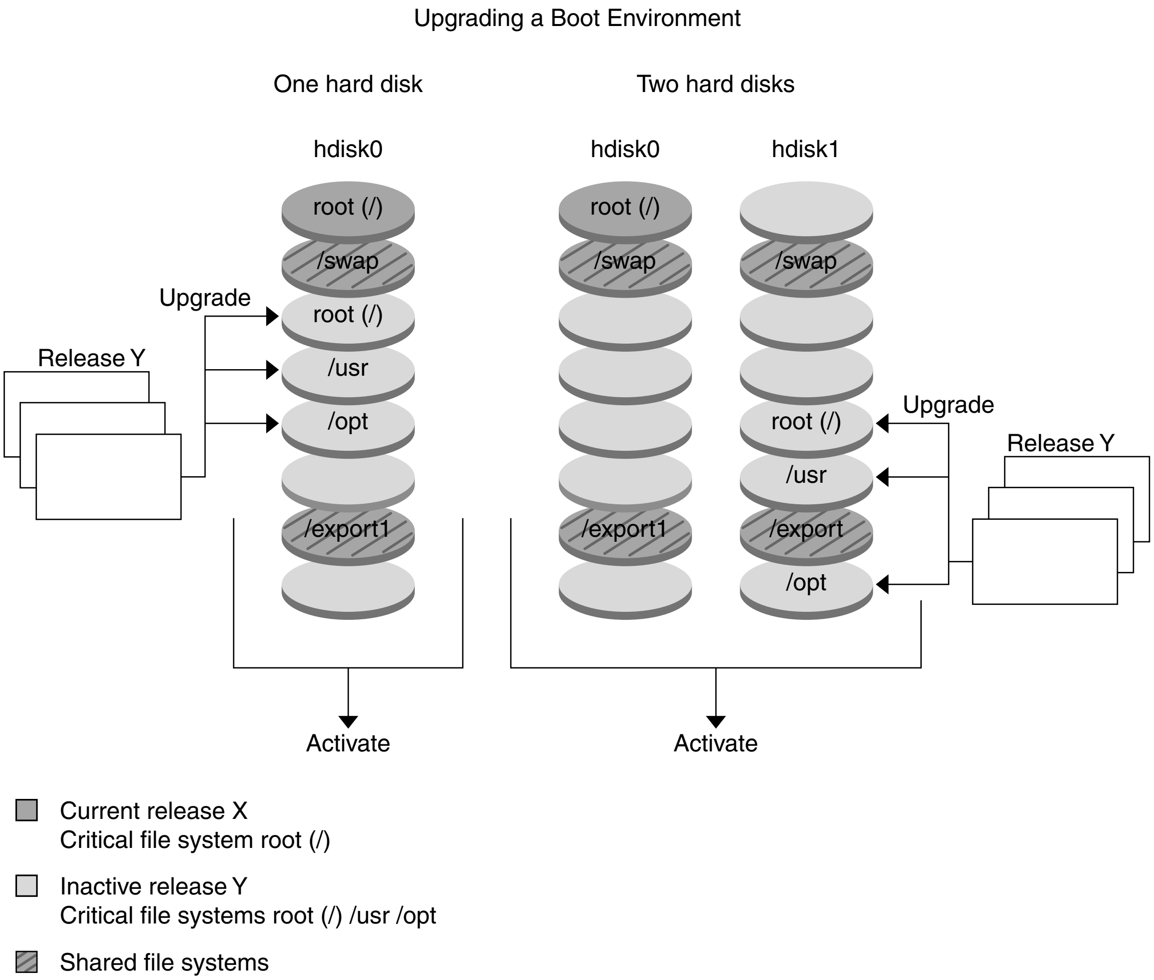


FIGURE 33–6 Upgrading an Inactive Boot Environment

Rather than an upgrade, you can install a Solaris Flash archive on a boot environment. The Solaris Flash installation feature enables you to create a single reference installation of the Solaris operating environment on a system. This system is called the master system. Then, you can replicate that installation on a number of systems that are called clone systems. In this situation, the inactive boot environment is a clone. When you install the Solaris Flash archive on a system, the archive replaces all the files on the existing boot environment as an initial installation would.

For procedures on installing a Solaris Flash archive, see “Installing Solaris Flash Archives on a Boot Environment” on page 456.

Figure 33–7 shows an installation of a Solaris Flash archive on an inactive boot environment.

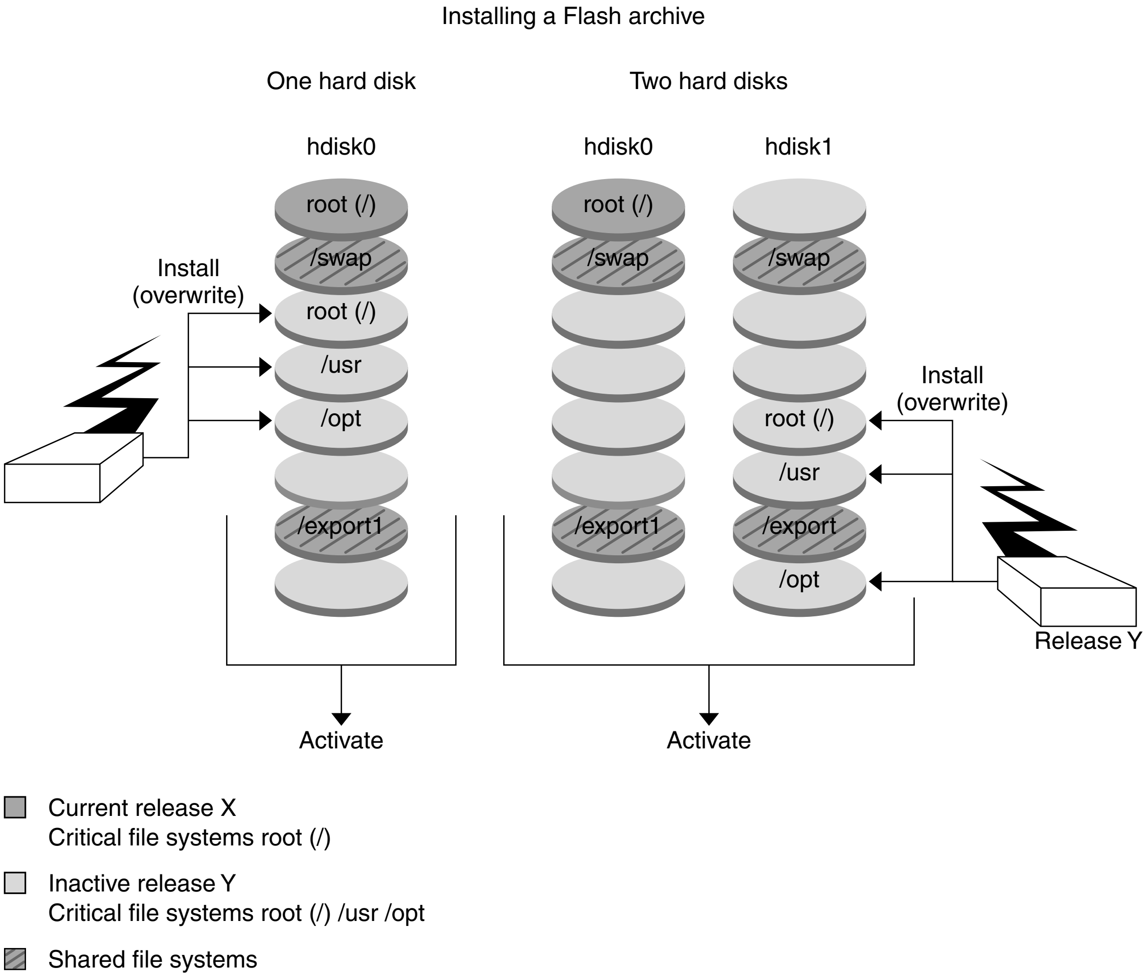


FIGURE 33–7 Installing a Solaris Flash Archive

Activating a Boot Environment

When you are ready to switch and make the new boot environment active, you quickly activate the new boot environment and reboot. Files are synchronized between boot environments the first time that you boot a newly created boot environment. "Synchronize" means that certain system files and directories are copied from the last-active boot environment to the boot environment being booted. When you reboot the system, the configuration that you installed on the new boot environment is active. The original boot environment then becomes an inactive boot environment.

- For procedures about activating a boot environment, see "Activating a Boot Environment" on page 461.
- For information about synchronizing the active and inactive boot environment, see "Synchronizing Files Between Boot Environments" on page 406.

Figure 33–8 shows a switch after a reboot from an inactive to an active boot environment.

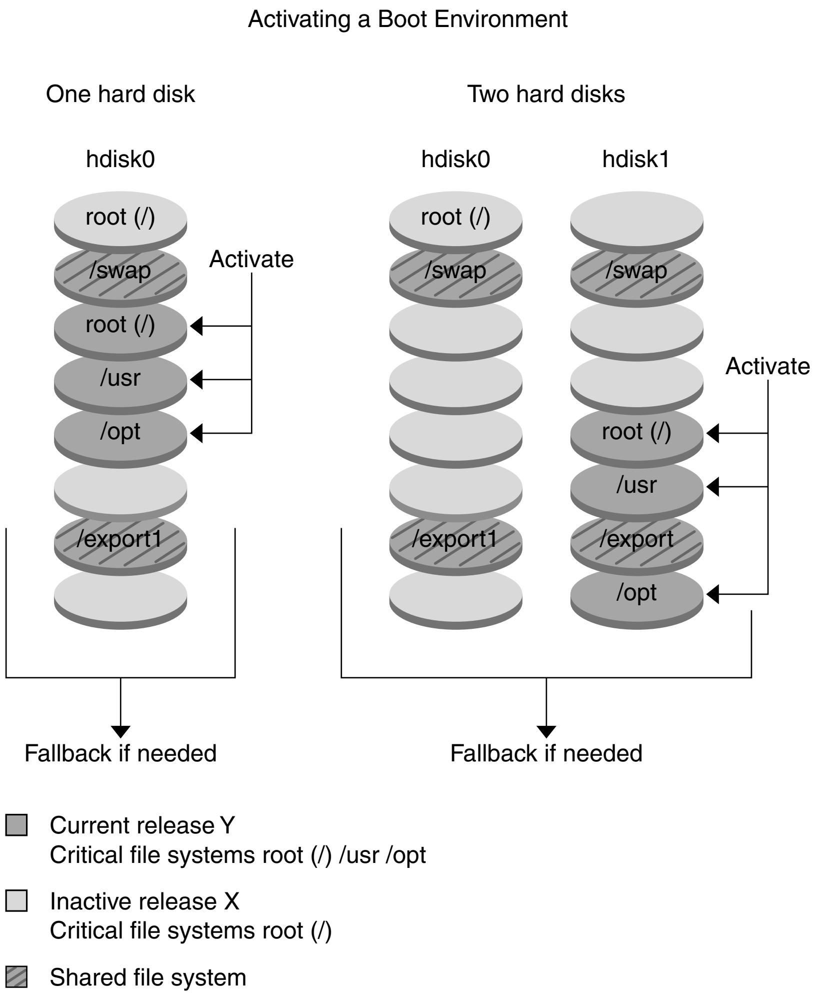


FIGURE 33–8 Activating an Inactive Boot Environment

Fall Back to the Original Boot Environment

If a failure occurs, you can quickly fall back to the original boot environment with an activation and reboot. You could fall back to the original boot environment for the following reasons:

- If the new boot environment cannot be booted
- If the new environment boots but does not work completely
- If you are not satisfied with the results

The use of fallback takes only the time to reboot the system, which is much quicker than backing up and restoring the original. The new boot environment that failed to boot is preserved. The failure can then be analyzed. You can only fall back to the boot environment that was used by `luactivate` to activate the new boot environment.

You fall back to the previous boot environment the following ways:

- If the new boot environment boots successfully, but you are not happy with the results, you run the `luactivate` command with the name of the previous boot environment and reboot.
- If the new boot environment does not boot, you boot the fallback boot environment in single-user mode and run the `luactivate` command and reboot.
- If you cannot boot in single-user mode, do one of the following:
  - Boot from DVD or CD media or a net installation image.
  - Mount the root (`/`) file system on the fallback boot environment.
  - Run the `luactivate` command and reboot.

For procedures to fall back, see “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 465.

Figure 33–9 shows the switch that is made when you reboot to fallback.

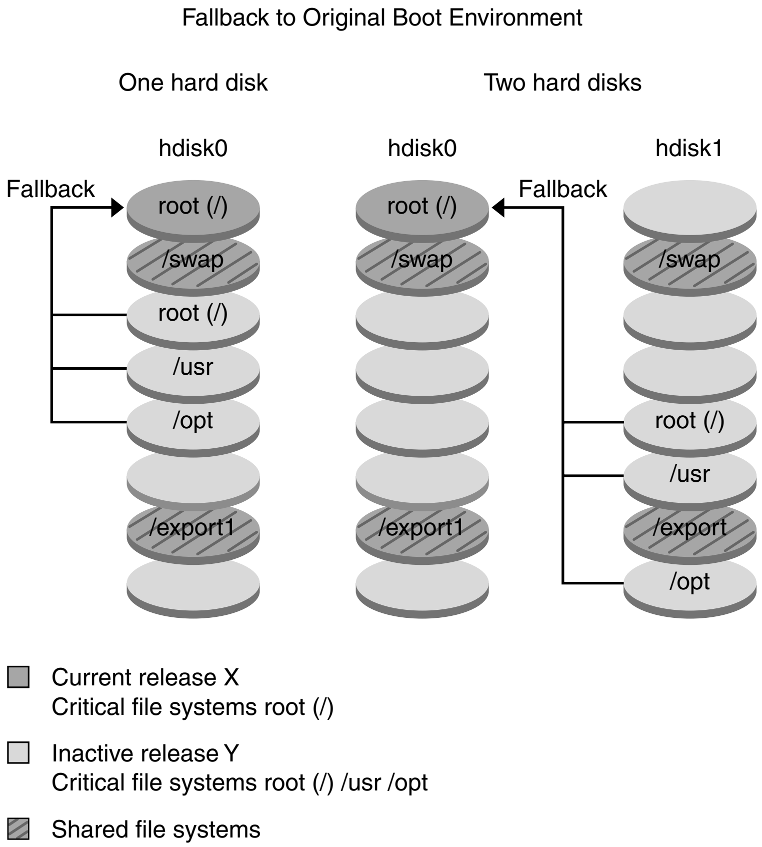


FIGURE 33–9 Fallback to the Original Boot Environment

Maintaining a Boot Environment

You can also do various maintenance activities such as checking status, renaming, or deleting a boot environment. For maintenance procedures, see Chapter 37.

CHAPTER **34**

Solaris Live Upgrade (Planning)

This chapter provides guidelines and requirements for review before installing and using Solaris Live Upgrade. You also should review general information on upgrading in "Checklist for Upgrading" on page 57. This chapter contains the following sections:

- "Solaris Live Upgrade Requirements" on page 395
- "Managing Packages and Patches With Solaris Live Upgrade" on page 398
- "Guidelines for Creating File Systems With the `lucreate` Command" on page 400
- "Guidelines for Selecting Slices for File Systems" on page 401
- "Customizing a New Boot Environment's Content" on page 405
- "Synchronizing Files Between Boot Environments" on page 406
- "Using Solaris Live Upgrade From a Remote System" on page 408

Solaris Live Upgrade Requirements

Solaris Live Upgrade System Requirements

Solaris Live Upgrade is included in the Solaris 9 software. If you want to upgrade by using Solaris Live Upgrade, you need to install the Solaris Live Upgrade packages on your current operating environment. You can upgrade a boot environment to a release of the Solaris Operating Environment that is the same as the release of the Solaris Live Upgrade packages installed on your machine. For example, if on your current Solaris 8 operating environment, you installed Solaris 9 Live Upgrade packages, you could upgrade a boot environment to the Solaris 9 marketing or update release.

Table 34–1 lists releases that are supported by Solaris Live Upgrade.

TABLE 34–1 Supported Solaris Releases

Platform	Release You Are Upgrading From	Release You Are Upgrading To
SPARC based system	Solaris 2.6, Solaris 7, or Solaris 8 operating environment	Solaris 8, operating environment
SPARC based system	Solaris 2.6, Solaris 7, or Solaris 8 operating environment	Solaris 9 operating environment
x86 based system	Solaris 7 operating environment	Solaris 8 operating environment
x86 based system	Solaris 7 or Solaris 8 operating environment	Solaris 9 operating environment

Note – You cannot upgrade to the Solaris 7 operating environment.

Installing Solaris Live Upgrade

You can install the Solaris Live Upgrade packages by using following:

- The `pkgadd` command. The Solaris Live Upgrade packages are `SUNWlur` and `SUNWluu`, and these packages must be installed in that order.
- An installer on the Solaris DVD, the Solaris Software 2 of 2 CD, or a net installation image.

Note – If you are running the Solaris 2.6, Solaris 7, or Solaris 8 release, you might not be able to run the Solaris Live Upgrade installer. These releases do not contain the set of patches needed to run the Java™ 2 runtime environment. You must have the Java 2 runtime environment recommended patch cluster to run the Solaris Live Upgrade installer and install the packages. To install the Solaris Live Upgrade packages, use the `pkgadd` command. Or, install the Java 2 runtime environment recommended patch cluster that is available on http://sunsolve.sun.com.

For instructions on installing the Solaris Live Upgrade software, see "Installing Solaris Live Upgrade" on page 411.

Solaris Live Upgrade Disk Space Requirements

Follow general disk space requirements for an upgrade. See Chapter 5.

To estimate the file system size that is needed to create a boot environment, start the creation of a new boot environment. The size is calculated. You can then abort the process.

The disk on the new boot environment must be able to serve as a boot device. Some systems restrict which disks can serve as a boot device. Refer to your system's documentation to determine if any boot restrictions apply.

The disk might need to be prepared before you create the new boot environment. Check to make sure the disk is formatted properly:

- Identify slices large enough to hold the file systems to be copied.
- Identify file systems that contain directories that you want to share between boot environments rather than copy. If you want a directory to be shared, you need to create a new boot environment with the directory put on its own slice. The directory is then a file system and can be shared with future boot environments. For more information on creating separate file systems for sharing, see "Guidelines for Selecting Slices for Shareable File Systems" on page 404.

Solaris Live Upgrade Requirements If Creating RAID-1 Volumes (Mirrors)

Solaris Live Upgrade uses Solaris Volume Manager technology to create a boot environment that can contain file systems that are RAID-1 volumes (mirrors). To use the mirroring capabilities of Solaris Live Upgrade, you must create at least one state database and at least three state database replicas. A state database stores information on disk about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a state database replica. When a state database is copied, the replica protects against data loss from single points of failure. For procedures about creating a state database, see "State Database (Overview)" in *Solaris Volume Manager Administration Guide*.

Solaris Live Upgrade does not implement the full functionality of Solaris Volume Manager. Solaris Live Upgrade supports only a RAID-1 volume (mirror) with single-slice concatenations on the root (/) file system. A mirror can be comprised of a maximum of three concatenations. For guidelines on creating mirrored file systems, see "Guidelines for Selecting Slices for Mirrored File Systems" on page 401.

Managing Packages and Patches With Solaris Live Upgrade

The following sections list packages required by Solaris Live Upgrade and provide information on recommended patches. See "Upgrading a System With Packages and Patches" on page 399 for information on using Solaris Live Upgrade to add packages and patches.

Caution – When upgrading and adding and removing packages or patches, Solaris Live Upgrade requires packages or patches that comply with the SVR4 Advanced Packaging Guidelines. While Sun packages conform to these guidelines, Sun cannot guarantee the conformance of packages from third-party vendors. If a package violates these guidelines, the package can cause the package-addition software during an upgrade to fail or alter the active boot environment.

For more information on adding and removing packages with Solaris Live Upgrade, see the man page, `luupgrade`(1M). For more information on packaging requirements, see Appendix G.

Required Packages

Check your current operating environment for the packages in the following table, which are required to use Solaris Live Upgrade. If packages in the column for your release are missing, use the `pkgadd` command to add the packages.

TABLE 34–2 Required Packages for Solaris Live Upgrade

Solaris 2.6 Release	Solaris 7 Release	Solaris 8 Release
SUNWadmap	SUNWadmap	SUNWadmap
SUNWadmc	SUNWadmc	SUNWadmc
SUNWjvrt	SUNWjvrt	SUNWj2rt
SUNWlibC	SUNWlibC	SUNWlibC
SUNWadmfw		SUNWbzip
SUNWmfrun		
SUNWloc		

To check for packages on your system, type the following command.

```
% pkginfo [package_name]
```

Upgrading a System With Packages and Patches

You can use Solaris Live Upgrade to add patches and packages to a system. By using Solaris Live Upgrade to add patches to a machine, the only downtime the system incurs is that of a reboot. You can add patches and packages to a boot environment with the `luupgrade` command or with a Solaris Flash archive.

- To add patches to a boot environment directly, create a new boot environment and use the `luupgrade` command with the `-t` option. To add packages to a boot environment, use the `luupgrade` command with the `-p` option. For more information, see the man page, `luupgrade`(1M).
- Or, you can use Solaris Live Upgrade to install a Solaris Flash archive. An archive contains a complete copy of a boot environment with new packages and patches already included. This complete boot environment or single reference system is called a master system. The process of creating a Solaris Flash archive begins with creating a master system. After you have created a master system, add any patches and packages that you want to install. Then, create a Solaris Flash archive of the master system. Use Solaris Live Upgrade to install the archive on the new boot environment. You can copy the boot environment and change and distribute the boot environment as many times as necessary. For details about how to create a Solaris Flash archive, see Chapter 21. For information on using Solaris Live Upgrade to install a Solaris Flash archive, see "Installing Solaris Flash Archives on a Boot Environment" on page 456.

Caution – When upgrading and adding and removing packages or patches, Solaris Live Upgrade requires packages or patches that comply with the SVR4 advanced packaging guidelines. While Sun packages conform to these guidelines, Sun cannot guarantee the conformance of packages from third-party vendors. If a package violates these guidelines, the package can cause the package-addition software to fail or can alter the active boot environment.

For more information on adding and removing packages with Solaris Live Upgrade, see the man page, `luupgrade`(1M). For more information on packaging requirements, see Appendix G.

Patches Needed to Run Solaris Live Upgrade

Correct operation of Solaris Live Upgrade requires that a limited set of patch revisions be installed for a given OS version. Before installing or running Live Upgrade, you are required to install a limited set of patch revisions. Make sure you have the most recently updated patch list by consulting http://sunsolve.sun.com. Search for the info doc 72099 on the SunSolve^{SM} web site.

Guidelines for Creating File Systems With the `lucreate` Command

The `lucreate -m` option specifies which file systems and the number of file systems to be created in the new boot environment You must specify the exact number of file systems you want to create by repeating this option. For example, a single use of the `-m` option specifies where to put all the file systems. You merge all the file systems from the original boot environment into the one file system specified by the `-m` option. If you specify the `-m` option twice, you create two file systems. When using the `-m` option to create file systems, follow these guidelines:

- You must specify one `-m` option for the root (/) file system for the new boot environment. If you run `lucreate` without the `-m` option, the Configuration menu is displayed. The Configuration menu enables you to customize the new boot environment by redirecting files onto new mount points.
- Any critical file systems that exist in the current boot environment and that are not specified in a `-m` option are merged into the next highest-level file system created.
- Only the file systems that are specified by the `-m` option are created on the new boot environment. If your current boot environment contains multiple file systems, and you want to have the same number of file systems in the new boot environment created, you must specify one `-m` option for each file system to be created. For example, if you have file systems for root (/), `/opt`, and `/var`, you would use one `-m` option for each file system on the new boot environment.
- Do not duplicate a mount point. For example, you cannot have two root (/) file systems.

Guidelines for Selecting Slices for File Systems

When you create file systems for a boot environment, the rules are identical to the rules for creating file systems for the Solaris operating environment. Solaris Live Upgrade cannot prevent you from creating invalid configurations for critical file systems. For example, you could type a `lucreate` command that would create separate file systems for root (/) and `/kernel`—an invalid division of root (/).

Do not overlap slices when re-slicing disks. If this condition exists, the new boot environment appears to have been created, but when activated, the boot environment does not boot. The overlapping file systems might be corrupted.

For Solaris Live Upgrade to work properly, the `vfstab` file on the active boot environment must have valid contents and must have an entry for root (/) at the minimum.

Guidelines for Selecting a Slice for the root (/) File System

When you create an inactive boot environment, you need to identify a slice where the root (/) file system is to be copied. Use the following guidelines when you select a slice for the root (/) file system. The slice must comply with the following:

- Must be a slice from which the system can boot.
- Must meet the recommended minimum size.
- If you have a sun4m system, the root (/) file system cannot be greater than 2 Gbytes.
- Can be on different physical disks or the same disk as the active root (/) file system.
- Can be a Veritas Volume Manager volume, but these volumes are not supported.

Guidelines for Selecting Slices for Mirrored File Systems

You can create a new boot environment that contains any combination of physical disk slices, Solaris Volume Manager volumes, or Veritas Volume Manager volumes. Critical file systems that are copied to the new boot environment can be of the following types:

- A physical slice.

- A single-slice concatenation that is included in a RAID–1 volume (mirror). The slice that contains the root (/) file system can be a RAID–1 volume.
- A single-slice concatenation that is included in a RAID–0 volume. The slice that contains the root (/) file system can be a RAID–0 volume.

When you create a new boot environment, the `lucreate -m` command recognizes the following three types of devices:

- A physical slice in the form of /dev/dsk/c*w*t*x*d*y*s*z*
- A Solaris Volume Manager volume in the form of /dev/md/dsk/d*num*
- A Veritas Volume Manager volume in the form of /dev/vx/dsk/*volume_name*

Note – If you have problems upgrading with Veritas VxVM, see "System Panics When Upgrading With Solaris Live Upgrade Running Veritas VxVm" on page 644.

General Guidelines for Creating Mirrored File Systems

Use the following guidelines to check if a RAID-1 volume is busy, resyncing, or if volumes contain file systems that are in use by a Solaris Live Upgrade boot environment.

For volume naming shortcuts and guidelines, see "Custom JumpStart and Solaris Live Upgrade Guidelines" on page 115.

Checking Status of Volumes

If a mirror or submirror needs maintenance or is busy, components cannot be detached. You should use the `metastat` command before creating a new boot environment and using the `detach` keyword. The `metastat` command checks if the mirror is in the process of resynchronization or if the mirror is in use. For information, see the man page `metastat`(1M).

Detaching Volumes and Resynchronizing Mirrors

If you use the `detach` keyword to detach a submirror, `lucreate` checks if a device is currently resyncing. If the device is resyncing, you cannot detach the submirror and you get an error message.

Resynchronization is the process of copying data from one submirror to another submirror after the following problems:

- Submirror failures.
- System crashes.
- A submirror has been taken offline and brought back online.

- The addition of a new submirror.

For more information about resynchronization, see “RAID 1 Volume (Mirror) Resynchronization” in *Solaris Volume Manager Administration Guide*.

Using Solaris Volume Manager Commands

Use the `lucreate` command rather than Solaris Volume Manager commands to manipulate volumes on inactive boot environments. The Solaris Volume Manager software has no knowledge of boot environments, whereas the `lucreate` command contains checks that prevent you from inadvertently destroying a boot environment. For example, `lucreate` prevents you from overwriting or deleting a Solaris Volume Manager volume.

However, if you have already used Solaris Volume Manager software to create complex Solaris Volume Manager concatenations, stripes, and mirrors, you must use Solaris Volume Manager software to manipulate them. Solaris Live Upgrade is aware of these components and supports their use. Before using Solaris Volume Manager commands that can create, modify, or destroy volume components, use the `lustatus` or `lufslist` commands. These commands can determine which Solaris Volume Manager volumes contain file systems that are in use by a Solaris Live Upgrade boot environment.

Guidelines for Selecting a Slice for a Swap File System

Configuring Swap for the New Boot Environment

You can configure a swap slice in three ways using the `lucreate` command with the `-m` option:

- If you do not specify a swap slice, the swap slices belonging to the current boot environment are configured for the new boot environment.
- If you specify one or more swap slices, these slices are the only swap slices that are used by the new boot environment. The two boot environments do not share any swap slices.
- You can specify to both share a swap slice and add a new slice for swap.

The following examples show the three ways of configuring swap. The current boot environment is configured with the root (/) file system on `c0t0d0s0`. The swap file system is on `c0t0d0s1`.

- In the following example, no swap slice is specified. The new boot environment contains the root (/) file system on `c0t1d0s0`. Swap is shared between the current and new boot environment on `c0t0d0s1`.

```
# lucreate -n be2 -m /:c0t1d0s0:ufs
```

- In the following example, a swap slice is specified. The new boot environment contains the root (/) file system on c0t1d0s0. A new swap file system is created on c0t1d0s1. No swap slice is shared between the current and new boot environment.

  ```
  # lucreate -n be2 -m /:c0t1d0s0:ufs -m -:c0t1d0s1:swap
  ```

- In the following example, a swap slice is added and another swap slice is shared between the two boot environments. The new boot environment contains the root (/) file system on c0t1d0s0. A new swap slice is created on c0t1d0s1. The swap slice on c0t0d0s1 is shared between the current and new boot environment.

  ```
  # lucreate -n be2 -m /:c0t1d0s0:ufs -m -:shared:swap -m -:c0t1d0s1:swap
  ```

Failed Boot Environment Creation If Swap Is In Use

A boot environment creation fails if the swap slice is being used by any boot environment except for the current boot environment. If the boot environment was created using the -s option, the alternate-source boot environment can be using the swap slice, but not any other boot environment.

Guidelines for Selecting Slices for Shareable File Systems

Solaris Live Upgrade copies the entire contents of a slice to the designated new boot environment slice. You might want some large file systems on that slice to be shared between boot environments rather than copied to conserve space and copying time. File systems that are critical to the operating environment such as root (/) and /var must be copied. File systems such as /home are not critical file systems and could be shared between boot environments. Shareable file systems must be user-defined file systems and on separate swap slices on both the active and new boot environments. You can reconfigure the disk several ways, depending your needs.

- You can reslice the disk before creating the new boot environment and put the shareable file system on its own slice. For example, if root (/), /var, and /home are on the same slice, reconfigure the disk and put /home on its own slice. When you create any new boot environments, /home is shared with the new boot environment by default.
- If you want to share a directory, the directory must be split off to its own slice. The directory is then a file system that can be shared with another boot environment. You can use the lucreate command with the -m option to create a new boot environment and split a directory off to its own slice. But, the new file system cannot yet be shared with the original boot environment. You need to run the lucreate command with the -m option again to create another boot environment. The two new boot environments can then share the directory.

For example, if you wanted to upgrade from the Solaris 8 release to the Solaris 9 release and share /home, you could run the lucreate command with the -m option. You could create a Solaris 8 release with /home as a separate file system on its own slice. Then run the lucreate command with the -m option again to duplicate that boot environment. This third boot environment can then be upgraded to the Solaris 9 release. /home is shared between the Solaris 8 and Solaris 9 releases.

For a description of shareable and critical file systems, see "File System Types" on page 378.

Customizing a New Boot Environment's Content

When you create a new boot environment, some directories and files can be excluded from a copy to the new boot environment. If you have excluded a directory, you can also re-include specified subdirectories or files under the excluded directory. These subdirectories or files that have been restored are then copied to the new boot environment. For example, you could exclude from the copy all files and directories in /etc/mail, but include all files and directories in /etc/mail/staff. The following command copies the staff subdirectory to the new boot environment.

```
# lucreate -n second_disk -x /etc/mail -y /etc/mail/staff
```

Caution – Use the file-exclusion options with caution. Do not remove files or directories that are required by the system.

The following table lists the lucreate command options for removing and restoring directories and files.

How Specified?	Options That Exclude	Options That Include
Specify the name of the directory or file	-x *exclude_dir*	-y *include_dir*
Use a file that contains a list	-f *list_filename*	-Y *list_filename*
	-z *list_filename*	-z *list_filename*

For examples of customizing the directories and files when creating a boot environment, see "To Create a Boot Environment and Customize the Content (Command-Line Interface)" on page 440.

Synchronizing Files Between Boot Environments

When you are ready to switch and make the new boot environment active, you quickly activate the new boot environment and reboot. Files are synchronized between boot environments the first time that you boot a newly created boot environment. “Synchronize” means that certain critical system files and directories might be copied from the last-active boot environment to the boot environment being booted. Those files and directories that have changed are copied.

Adding Files to the `/etc/lu/synclist`

Solaris Live Upgrade checks for critical files that have changed. If these files' content is not the same in both boot environments, they are copied from the active boot environment to the new boot environment. Synchronizing is meant for critical files such as `/etc/passwd` or `/etc/group` files that might have changed since the new boot environment was created.

The `/etc/lu/synclist` file contains a list of directories and files that are synchronized. In some instances, you might want to copy other files from the active boot environment to the new boot environment. You can add directories and files to `/etc/lu/synclist` if necessary.

Adding files not listed in the `/etc/lu/synclist` could cause a system to become unbootable. The synchronization process only copies files and creates directories. The process does not remove files and directories.

The following example of the `/etc/lu/synclist` file shows the standard directories and files that are synchronized for this system.

```
/var/mail                    OVERWRITE
/var/spool/mqueue            OVERWRITE
/var/spool/cron/crontabs     OVERWRITE
/var/dhcp                    OVERWRITE
/etc/passwd                  OVERWRITE
/etc/shadow                  OVERWRITE
/etc/opasswd                 OVERWRITE
/etc/oshadow                 OVERWRITE
/etc/group                   OVERWRITE
/etc/pwhist                  OVERWRITE
/etc/default/passwd          OVERWRITE
/etc/dfs                     OVERWRITE
/var/log/syslog              APPEND
/var/adm/messages            APPEND
```

Examples of directories and files that might be appropriate to add to the synclist file are the following:

```
/var/yp                     OVERWRITE
/etc/mail                   OVERWRITE
/etc/resolv.conf            OVERWRITE
/etc/domainname             OVERWRITE
```

The synclist file entries can be files or directories. The second field is the method of updating that occurs on the activation of the boot environment. There are three methods for updating the files:

- OVERWRITE — The contents of the active boot environment's file overwrites the contents of the new boot environment file. OVERWRITE is the default action if no action is specified in the second field. If the entry is a directory, all subdirectories are copied. All files are overwritten. The new boot environment file has the same date, mode, and ownership as the same file on the previous boot environment.
- APPEND — The contents of the active boot environment's file are added to the end of the new boot environment's file. This might lead to duplicate entries in the file. Directories cannot be listed as APPEND. The new boot environment file has the same date, mode, and ownership as the same file on the previous boot environment.
- PREPEND — The contents of the active boot environment's file are added to the beginning of the new boot environment's file. This might lead to duplicate entries in the file. Directories can not be listed as PREPEND. The new boot environment file has the same date, mode, and ownership as the same file on the previous boot environment.

Forcing a Synchronization Between Boot Environments

The first time you boot from a newly created boot environment, Solaris Live Upgrade synchronizes the new boot environment with the boot environment that was last active. After this initial boot and synchronization, Solaris Live Upgrade does not perform a synchronization unless requested.

- To force synchronization using the CUI, you type **yes** when prompted
- To force synchronization using the CLI, you use the luactivate command with the -s option

You might want to force a synchronization if you are maintaining multiple versions of the Solaris operating environment. You might want changes in files such as email or passwd/group to be in the boot environment you are activating to. If you force a synchronization, Solaris Live Upgrade checks for conflicts between files that are subject to synchronization. When the new boot environment is booted and a conflict is detected, a warning is issued and the files are not synchronized. Activation can be completed successfully, in spite of such a conflict. A conflict can occur if you make changes to the same file on both the new boot environment and the active boot environment. For example, you make changes to the /etc/passwd file on the original boot environment. Then you make other changes to /etc/passwd file on the new boot environment. The synchronization process cannot choose which file to copy for the synchronization.

⚠ **Caution –** Use this option with great care, because you might not be aware or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 9 software on your current boot environment and booted back to a Solaris 7 release with a forced synchronization, files could be changed on the 7 release. Because files are dependent on the release of the operating environment, the boot to the Solaris 7 release could fail because the Solaris 9 files might not be compatible with the Solaris 7 files.

Using Solaris Live Upgrade From a Remote System

When viewing the character interface remotely, such as over a tip line, you might need to set the TERM environment variable to VT220. Also, when using the Common Desktop Environment (CDE), set the value of the TERM variable to dtterm, rather than xterm.

CHAPTER 35

Using Solaris Live Upgrade to Create a Boot Environment (Tasks)

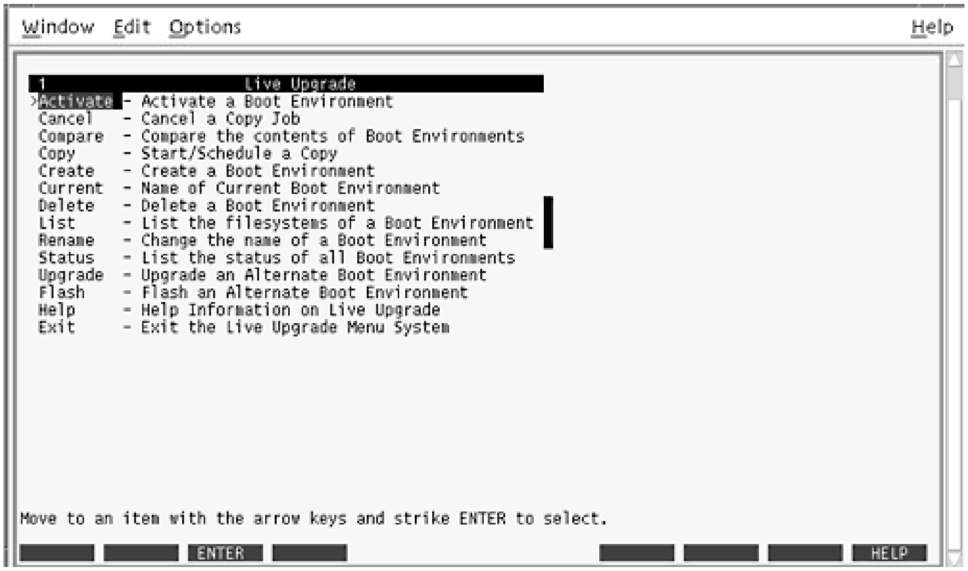
This chapter explains how to install Solaris Live Upgrade, use the menus, and to create a boot environment. This chapter contains the following sections:

About Solaris Live Upgrade Interfaces

You can run Solaris Live Upgrade with a character user interface (CUI) or the command-line interface (CLI). Procedures for both the CUI and CLI are provided in the following sections. The CUI does not provide access to all features of Solaris Live Upgrade. The CLI procedures in this document cover the basic uses of the Solaris Live Upgrade commands. See Chapter 39 for a list of commands and also see the appropriate, associated man pages, for more options to use with these commands.

The CUI does not run in multibyte locales and 8–bit locales.

Using Solaris Live Upgrade Menus



```
1                      Live Upgrade
>Activate - Activate a Boot Environment
 Cancel   - Cancel a Copy Job
 Compare  - Compare the contents of Boot Environments
 Copy     - Start/Schedule a Copy
 Create   - Create a Boot Environment
 Current  - Name of Current Boot Environment
 Delete   - Delete a Boot Environment
 List     - List the filesystems of a Boot Environment
 Rename   - Change the name of a Boot Environment
 Status   - List the status of all Boot Environments
 Upgrade  - Upgrade an Alternate Boot Environment
 Flash    - Flash an Alternate Boot Environment
 Help     - Help Information on Live Upgrade
 Exit     - Exit the Live Upgrade Menu System

Move to an item with the arrow keys and strike ENTER to select.
```

FIGURE 35–1 Solaris Live Upgrade Main Menu

Navigation through the menus of the Solaris Live Upgrade character user interface requires that you use arrow keys and function keys. Use arrow keys to navigate up and down before making a selection or to place the cursor in a field. To perform a task, use the function keys. At the bottom of the menu, you see black rectangles that represent function keys on the keyboard. The first black rectangle represents F1, the second F2, and so on. Rectangles that are active contain a word that represents a task such as Save. The Configuration menu notes the function key number plus the task, rather than a rectangle.

- F3 is always SAVE and completes the task for that menu.
- F6 is always CANCEL and exits the menu without saving changes.
- Other function keys’ tasks vary, depending on the menu.

In the following procedures, you might be asked to press a function key. If your function keys do not properly map to the function keys on the Solaris Live Upgrade menus, use Control-F plus the appropriate number.

Task Map: Installing Solaris Live Upgrade and Creating Boot Environments

TABLE 35–1 Task Map: Using Solaris Live Upgrade

Task	Description	For Instructions
Install Solaris Live Upgrade packages	Install packages on your operating environment	"Installing Solaris Live Upgrade" on page 411
Start Solaris Live Upgrade	Start the Solaris Live Upgrade main menu	"Starting and Stopping Solaris Live Upgrade (Character Interface)" on page 413
Create a boot environment	Copy and reconfigure file systems to an inactive boot environment	"Creating a New Boot Environment" on page 414

Installing Solaris Live Upgrade

You need to install the Solaris Live Upgrade packages on your current operating environment. The release of the Solaris Live Upgrade packages must match the release of the Operating Environment you are upgrading to. For example, if your current operating environment is the Solaris 8 release and you want to upgrade to the Solaris 9 release, you need to install the Solaris Live Upgrade packages from the Solaris 9 release.

You can install packages using the `pkgadd` command or using the Solaris Live Upgrade installer. See "To Install Solaris Live Upgrade With the `pkgadd` Command" on page 412 or "To Install Solaris Live Upgrade With the Solaris Web Start Installer" on page 412.

Note – If you are running theSolaris 2.6, Solaris 7, or Solaris 8 release, you might not be able to run the Solaris Live Upgrade installer. These releases do not contain the set of patches needed to run the Java 2 runtime environment. You must have the Java 2 runtime environment recommended patch cluster to run the Solaris Live Upgrade installer and install the packages. To install the Solaris Live Upgrade packages, use the `pkgadd` command. Or, install the Java 2 runtime environment recommended patch cluster that is available on http://sunsolve.sun.com.

▼ To Install Solaris Live Upgrade With the `pkgadd` Command

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

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

2. **Install the packages in the following order.**

   ```
   # pkgadd -d path_to_packages SUNWlur SUNWluu
   ```

   path_to_packages   Specifies the absolute path to the software packages.

3. **Verify that the package has been installed successfully.**

   ```
   # pkgchk -v SUNWlur SUNWluu
   ```

▼ To Install Solaris Live Upgrade With the Solaris Web Start Installer

1. **Insert the Solaris DVD or Solaris Software 2 of 2 CD.**

2. **Run the installer for the media you are using.**

   - If you are using the Solaris DVD, change directories to the installer and run the installer.

     ```
     # cd /cdrom/cdrom0/Solaris_9/Tools/installers
     # ./liveupgrade20
     ```

     The Solaris Web Start installer is displayed.

   - If you are using the Solaris Software 2 of 2 CD, run the installer.

     ```
     % ./installer
     ```

     The Solaris Web Start installer is displayed.

3. **From the Select Type of Install panel, click Custom.**
4. **On the Locale Selection panel, click the language to be installed.**
5. **Choose the software to install.**
   - For DVD, on the Component Selection panel, click Next to install the packages.
   - For CD, on the Product Selection panel, click Default Install for Solaris Live Upgrade and click on the other software choices to deselect them.
6. **Follow the directions on the Solaris Web Start installer panels to install the software.**

Starting and Stopping Solaris Live Upgrade (Character Interface)

This procedure starts and stops the Solaris Live Upgrade menu program.

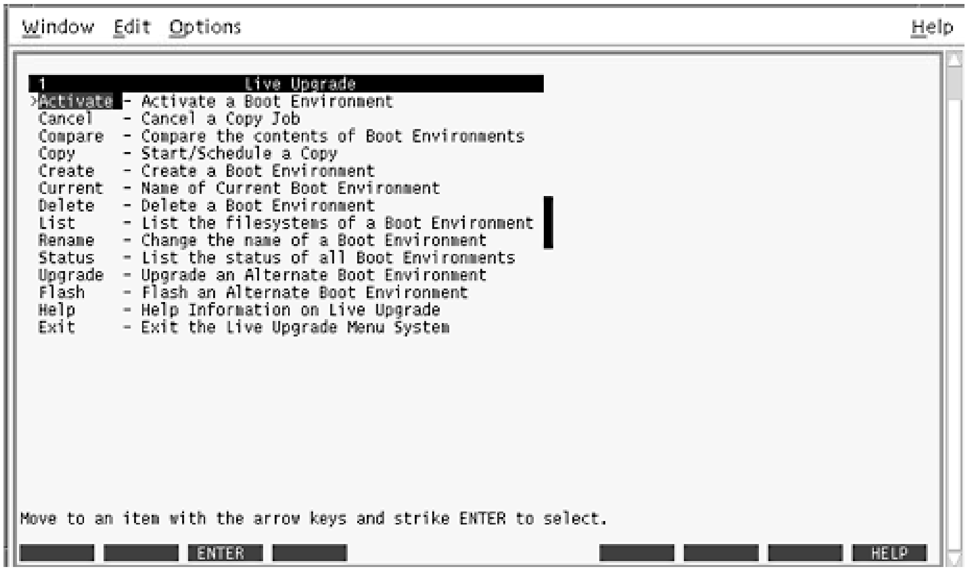
▼ To Start Solaris Live Upgrade Menus

Note – When viewing the character interface remotely, such as over a tip line, you might need to set the `TERM` environment variable to VT220. Also, when using the Common Desktop Environment (CDE), set the value of the `TERM` variable to `dtterm`, rather than `xterm`.

1. **Log in as superuser.**
2. **Type:**

   ```
   # /usr/sbin/lu
   ```

   The Solaris Live Upgrade main menu is displayed.



```
1                    Live Upgrade
>Activate - Activate a Boot Environment
 Cancel   - Cancel a Copy Job
 Compare  - Compare the contents of Boot Environments
 Copy     - Start/Schedule a Copy
 Create   - Create a Boot Environment
 Current  - Name of Current Boot Environment
 Delete   - Delete a Boot Environment
 List     - List the filesystems of a Boot Environment
 Rename   - Change the name of a Boot Environment
 Status   - List the status of all Boot Environments
 Upgrade  - Upgrade an Alternate Boot Environment
 Flash    - Flash an Alternate Boot Environment
 Help     - Help Information on Live Upgrade
 Exit     - Exit the Live Upgrade Menu System

Move to an item with the arrow keys and strike ENTER to select.
```

FIGURE 35–2 Solaris Live Upgrade Main Menu

▼ To Stop Solaris Live Upgrade Menus

● **From the main menu, select Exit.**

Creating a New Boot Environment

Creating a boot environment provides a method of copying critical file systems from the active boot environment to a new boot environment. The CUI's Create menu and Configuration submenu, and the `lucreate` command enable reorganizing a disk if necessary, customizing file systems, and copying the critical file systems to the new boot environment.

Before file systems are copied to the new boot environment, they can be customized so that critical file system directories are either merged into their parent directory or split from their parent directory. User-defined (shareable) file systems are shared between boot environments by default. But shareable file systems can be copied if needed. Swap, which is a shareable file system, can be split and merged also. For an overview of critical and shareable file systems, see "File System Types" on page 378.

▼ To Create a Boot Environment (Character Interface)

1. **From the main menu, select Create.**

   The system displays the Create a Boot Environment submenu.

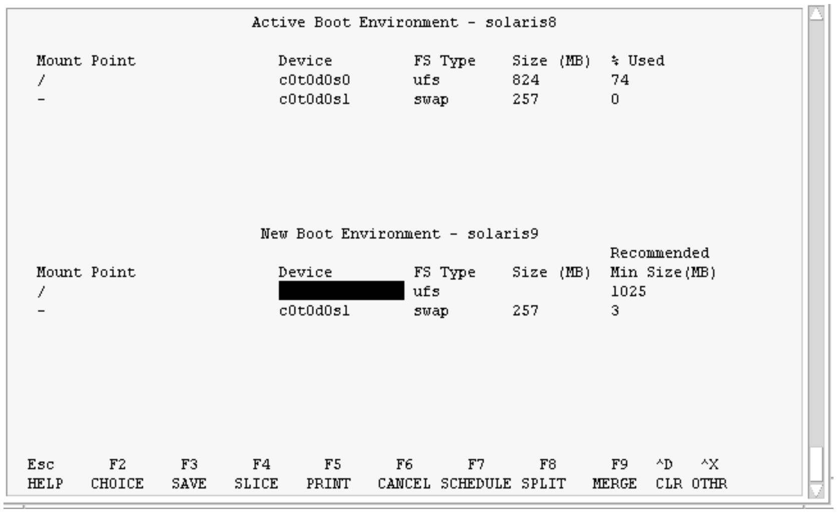
2. **Type the name of the active boot environment (if necessary) and the new boot environment and confirm. You are only required to type the name of the active boot environment the first time you create a boot environment.**

   The boot environment name can be no longer than 30 characters, can contain only alphanumeric characters, and can contain no multibyte characters.

   ```
   Name of Current Boot Environment:    solaris8
   Name of New Boot Environment:   solaris9
   ```

3. **To save your changes, press F3.**

   The configuration menu appears.



```
                          Active Boot Environment - solaris8

  Mount Point                 Device        FS Type    Size (MB)  % Used
  /                           c0t0d0s0      ufs        824        74
  -                           c0t0d0s1      swap       257        0




                           New Boot Environment - solaris9
                                                                  Recommended
  Mount Point                 Device        FS Type    Size (MB)  Min Size(MB)
  /                           [redacted]    ufs                   1025
  -                           c0t0d0s1      swap       257        3




 Esc      F2      F3      F4      F5      F6      F7       F8      F9   ^D   ^X
 HELP   CHOICE   SAVE   SLICE   PRINT   CANCEL SCHEDULE SPLIT   MERGE  CLR OTHR
```

FIGURE 35–3 Solaris Live Upgrade Configuration Menu

The Configuration menu contains the following parts:

- The original boot environment is located at the top of the screen. The boot environment to be created is at the bottom.
- The Device field contains the following information.
  - The name of a disk device is in the form /dev/dsk/c*w*t*x*d*y*s*z*.
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/d*num*.

    - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/*volume_name*.
    - The area for selecting a critical file system is blank until you select a critical file system. The critical file systems such as `/usr`, `/var`, or `/opt` can be split or merged with the root (`/`) file system.
    - Shareable file systems such as `/export` or swap are displayed in the Device field. These file systems contain the same mount point in both the source and target boot environments. Swap is shared by default, but you can also split and merge (add and remove) swap slices.

      For an overview of critical and shareable file systems, see "File System Types" on page 378.
  - The FS_Type field enables you to change file system type. The file system type can be one of the following:
    - `vxfs`, which indicates a Veritas file system
    - `swap`, which indicates a swap file system
    - `ufs`, which indicates a UFS file system

4. **(Optional) The following tasks can be done at any time:**
   - To print the information on-screen to an ASCII file, press F5.
   - To scroll through the file system list, press Control-X.

     You can then switch between the file systems of the active and new boot environment and scroll.
   - To exit the Configuration menu at any time, press F6.
     - If you are in the Configuration menu, changes are not saved and file systems are not altered.
     - If you are in a Configuration submenu, you return to the Configuration menu.

5. **Select an available slice by pressing F2.**

   The Choices menu displays available slices on the system for the field where the cursor is placed. The menu displays a Slice field and a file system FS_Type field.

   a. **Use the arrow keys to place the cursor in a field to select a slice or file system type.**
      - When you place your cursor in the Slice field, all free slices are displayed. For root (`/`), Choices only displays free slices that meet the root (`/`) file system limitations. See "Guidelines for Selecting a Slice for the root (`/`) File System" on page 401.
      - When you place your cursor in the FS_Type field, all available file system types are displayed.

- Slices in bold can be selected for the current file system. The size of the slice is estimated by adding the size of the file system plus 30 percent to accommodate an upgrade.
- Slices not in bold are too small to support the given file system. To reslice a disk, see Step 6.

b. **Press Return to choose a slice.**

The slice appears in the Slice field or the file system type changes in the FS_Type field.

6. **(Optional) If available slices do not meet the minimum requirements, to reslice any available disks, press F4.**

The Solaris Live Upgrade Slice Configuration menu appears.

The `format` command runs, which enables you to create new slices. Follow the screen to create a new slice. For information on the `format` command, see the `format` man page.

To navigate through this menu, use the arrow keys to move between the Device field and FS_Type field. The Size (Mbytes) field is automatically completed as the devices are selected.

a. **To free a device, press Control-D.**

The slice is now available and appears on the Choices menu.

b. **To return to the Configuration menu, press F3.**

7. **(Optional) Splitting critical file systems puts the file systems on separate mount points. To split a file system, do the following:**

(To merge file systems, see Step 8).

a. **Select the file system to split.**

You can split or exclude file systems such as `/usr`, `/var`, or `/opt` from their parent directory.

Note – When creating file systems for a boot environment, the rules are identical to the rules for creating file systems for the Solaris operating environment. Solaris Live Upgrade cannot prevent you from making invalid configurations on critical file systems. For example, you could enter a `lucreate` command that would create separate file systems for root (/) and `/kernel`—an invalid division of root (/).

---

b. **Press F8.**

c. **Type the file system name for the new boot environment, for example:**

```
Enter the directory that will be a separate file system
on the new boot environment: /opt
```

When the new file system is verified, a new line is added to the screen.

d. **To return to the Configuration menu, press F3.**

The Configuration menu is displayed.

8. **(Optional) Merging puts the file systems on the same mount point. To merge a file system into its parent directory:**

(To split file systems, see Step 7.)

a. **Select the file system to merge.**

You can merge file systems such as `/usr`, `/var`, or `/opt` into their parent directory.

b. **Press F9.**

The file systems that will be combined are displayed, for example:

```
/opt will be merged into /.
```

c. **Press Return.**

d. **To return to the Configuration menu, press F3.**

The Configuration menu is displayed.

9. **(Optional) Decide if you want to add or remove swap slices.**

- If you want to split a swap slice and put swap on a new slice, continue with Step 10.
- If you want to remove a swap slice, continue with Step 11.

10. **(Optional) To split a swap slice, do the following:**

a. **In the Device field, select the swap slice that you want to split.**

b. **Press F8.**

c. **At the prompt, type:**

```
Enter the directory that will be a separate filesystem on
the new BE: swap
```

d. **Press F2 Choice.**

The Choice menu lists the available slices for swap.

e. **Select the slice to put swap on.**

The slice appears in the Device field and you have a new slice for swap.

11. **(Optional) To remove a swap slice, do the following:**

a. **In the Device field, select the swap slice that you are removing.**

b. **Press F9.**

c. **At the prompt, type y.**

```
Slice /dev/dsk/c0t4d0s0 will not be swap partition.
Please confirm? [y, n]: y
```

The swap slice no longer exists.

12. **Decide if you want to create the boot environment now or schedule the creation for later:**

- Press F3 to create the new boot environment now.

  The configuration is saved and you exit the configuration screen. The file systems are copied, the boot environment is made bootable, and an inactive boot environment is created.

  Creating a boot environment might take an hour or more, depending on your system configuration. The Solaris Live Upgrade main menu is then displayed.

- If you want to schedule the creation for a later time, type **y**, then the start time, and an email address, as in this example.

```
Do you want to schedule the copy? y
Enter the time in 'at' format to schedule create: 8:15 PM
Enter the address to which the copy log should be mailed: someone@anywhere.com
```

  You are notified of the completion by email.

  For information on time formats, see the `at`(1) man page.

  You can schedule only one job at a time.

After the creation is complete, the inactive boot environment is ready to be upgraded. See Chapter 36.

▼ To Create a Boot Environment for the First Time (Command-Line Interface)

The `lucreate` command that is used with the `-m` option specifies which file systems and the number of file systems to be created in the new boot environment. You must specify the exact number of file systems you want to create by repeating this option. For example, a single use of the `-m` option specifies where to put all the file systems. You merge all the file systems from the original boot environment into the one file system specified by the `-m` option. If you specify the `-m` option twice, you create two file systems. When using the `-m` option to create file systems, follow these guidelines:

- You must specify one -m option for the root (/) file system for the new boot environment. If you run lucreate without the -m option, the Configuration menu is displayed. The Configuration menu enables you to customize the new boot environment by redirecting files onto new mount points.
- Any critical file systems that exist in the current boot environment and are not specified in a -m option are merged into the next highest-level file system created.
- Only the file systems that are specified by the -m option are created on the new boot environment. If your current boot environment contains multiple file systems, and you want to have the same number of file systems in the new boot environment created, you must specify one -m option for each file system to be created. For example, if you have file systems for root (/), /opt, and /var, you would use one -m option for each file system on the new boot environment.
- Do not duplicate a mount point. For example, you cannot have two root (/) file systems.

1. **Log in as superuser.**
2. **To create the new boot environment, type:**

```
# lucreate [-A 'BE_description'] -c BE_name \
-m mountpoint:device[,metadevice]:fs_options [-m ...] -n BE_name
```

-A *'BE_description'*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

-c *BE_name*
Assigns the name *BE_name* to the active boot environment. This option is not required and is only used when the first boot environment is created. If you run lucreate for the first time and you omit the -c option, the software creates a default name for you.

The default name is chosen according to the following criteria:

- If the physical boot device can be determined, then the base name of the physical boot device is used to name the current boot environment.

  For example, if the physical boot device is /dev/dsk/c0t0d0s0, then the current boot environment is given the name c0t0d0s0.
- If the physical boot device cannot be determined, then names from the uname command with the -s and -r options are combined to produce the name.

  For example, if the uname -s returns the operating environment name of SunOS and the uname -r returns the release name of 5.9, then the name SunOS5.9 is given to the current boot environment.
- If both of the above cannot determine the name, then the name current is used to name the current boot environment.

Note – If you use the -c option after the first boot environment creation, the option is ignored or an error message is displayed.

- If the name specified is the same as the current boot environment name, the option is ignored.
- If the name specified is different than the current boot environment name, then an error message is displayed and the creation fails. The following example shows a boot environment name that causes an error message.

```
# lucurr
c0t0d0s0
# lucreate -c c1t1d1s1 -n newbe -m /:c1t1d1s1:ufs
ERROR: current boot environment name is c0t0d0s0: cannot change
name using <-c c1t1d1s1>
```

-m *mountpoint:device[,metadevice]:fs_options* [-m ...]

Specifies the file systems' configuration of the new boot environment in the vfstab. The file systems that are specified as arguments to -m can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
  - The name of a disk device, of the form /dev/dsk/c*w*t*x*d*y*s*z*
  - The name of a Solaris Volume Manager volume, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/md/vxfs/dsk/d*num*
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
  - ufs, which indicates a UFS file system.
  - vxfs, which indicates a Veritas file system.
  - swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
  - For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

-n *BE_name*

The name of the boot environment to be created. *BE_name* must be unique on the system.

EXAMPLE 35–1 Creating a Boot Environment (Command Line)

In this example, the active boot environment is named first_disk. The mount points for the file systems are noted by using the -m option. Two file systems are created, root (/) and /usr. The new boot environment is named second_disk. A description, mydescription, is associated with the name second_disk. Swap, in the new boot environment second_disk, is automatically shared from the source, first_disk.

```
# lucreate -A 'mydescription' -c first_disk
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \
-n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

▼ To Create a Boot Environment and Merge File Systems (Command-Line Interface)

Note – The lucreate command used with the -m option specifies which file systems and the number of file systems to be created in the new boot environment. You must specify the exact number of file systems you want to create by repeating this option. For example, a single use of the -m option specifies where to put all the file systems. You merge all the file systems from the original boot environment into one file system. If you specify the -m option twice, you create two file systems.

1. **Log in as superuser.**
2. **Type:**

```
# lucreate  [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
[-m [...]] -m mountpoint:merged:fs_options -n BE_name
```

-A *BE_description*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

-m *mountpoint:device[,metadevice]:fs_options* [-m...]
Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to -m can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
  - The name of a disk device, of the form /dev/dsk/c*w*t*x*d*y*s*z*
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/*volume_name*
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
  - ufs, which indicates a UFS file system.
  - vxfs, which indicates a Veritas file system.
  - swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
  - For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

-n *BE_name*

The name of the boot environment to be created. *BE_name* must be unique on the system.

EXAMPLE 35–2 Creating a Boot Environment and Merge File Systems (Command-Line Interface)

In this example, the file systems on the current boot environment are root (/), /usr, and /opt. The /opt file system is combined with its parent file system /usr. The new boot environment is named second_disk. A description, mydescription, is associated with the name second_disk.

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
-m /usr/opt:merged:ufs -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 36.

▼ To Create a Boot Environment and Split File Systems (Command-Line Interface)

Note – When creating file systems for a boot environment, the rules are identical to the rules for creating file systems for the Solaris operating environment. Solaris Live Upgrade cannot prevent you from making invalid configurations on critical file systems. For example, you could enter an `lucreate` command that would create separate file systems for root (/) and `/kernel`—an invalid division of root (/).

When splitting a directory into multiple mount points, hard links are not maintained across file systems. For example, if `/usr/stuff1/file` is hard linked to `/usr/stuff2/file`, and `/usr/stuff1` and `/usr/stuff2` are split into separate file systems, the link between the files no longer exists. `lucreate` issues a warning message and a symbolic link is created to replace the lost hard link.

1. **Log in as superuser.**
2. **Type:**

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
[-m mountpoint:device[,metadevice]:fs_options] -n new_BE
```

`-A` *'BE_description'*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and contain any characters.

`-m` *mountpoint:device[,metadevice]:fs_options* [`-m`...]
Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
  - The name of a disk device, of the form `/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z*
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/*volume_name*
  - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:

- ufs, which indicates a UFS file system.
- vxfs, which indicates a Veritas file system.
- swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
- For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

-n *BE_name*

The name of the boot environment to be created. *BE_name* must be unique on the system.

EXAMPLE 35–3 Creating a Boot Environment and Splitting File Systems (Command-Line Interface)

In this example, the preceding command splits the root (/) file system over multiple disk slices in the new boot environment. Assume a source boot environment that has /usr, /var, and /opt on root (/): /dev/dsk/c0t0d0s0 /.

On the new boot environment, separate /usr, /var, and /opt, mounting these file systems on their own slices, as follows:

```
/dev/dsk/c0t1d0s0 /

/dev/dsk/c0t1d0s1 /var

/dev/dsk/c0t1d0s7 /usr

/dev/dsk/c0t1d0s5 /opt
```

A description, mydescription, is associated with the boot environment name second_disk.

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t1d0s0:ufs -m /usr:/dev/dsk/c0t1d0s7:ufs  \
-m /var:/dev/dsk/c0t1d0s1:ufs -m /opt:/dev/dsk/c0t1d0s5:ufs \
-n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 36.

▼ To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)

Swap slices are shared between boot environments by default. By *not* specifying swap with the `-m` option, your current and new boot environment share the same swap slices. If you want to reconfigure the new boot environment's swap, use the `-m` option to add or remove swap slices in the new boot environment.

Note – The swap slice cannot be in use by any boot environment except the current boot environment or if the `-s` option is used, the source boot environment. The boot environment creation fails if the swap slice is being used by any other boot environment, whether it is a swap, UFS, or any other file system.

You can create a boot environment with the existing swap slices and then edit the `vfstab` file after the creation.

---

1. **Log in as superuser.**
2. **Type:**

```
# lucreate  [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
-m -:device:swap -n BE_name
```

`-A` *'BE_description'*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

`-m` *mountpoint:device[,metadevice]:fs_options* [`-m`...]
Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
  - The name of a disk device, of the form `/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z*
  - The name of a Solaris Volume Manager metadevice, of the form `/dev/md/dsk/d`*num*
  - The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/`*volume_name*
  - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:

- ufs, which indicates a UFS file system.
- vxfs, which indicates a Veritas file system.
- swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
- For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see “To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)” on page 435.

-n *BE_name*
The name of the boot environment to be created. *BE_name* must be unique.

The new boot environment is created with swap moved to a different slice or device.

EXAMPLE 35–4 Creating a Boot Environment and Reconfigure Swap (Command-Line Interface)

In this example, the current boot environment contains root (/) on /dev/dsk/c0t0d0s0 and swap is on /dev/dsk/c0t0d0s1. The new boot environment copies root (/) to /dev/dsk/c0t4d0s0 and uses both /dev/dsk/c0t0d0s1 and /dev/dsk/c0t4d0s1 as swap slices. A description, mydescription, is associated with the boot environment name second_disk.

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m -:/dev/dsk/c0t0d0s1:swap \
-m -:/dev/dsk/c0t4d0s1:swap -n second_disk
```

These swap assignments are effective only after booting from second_disk. If you have a long list of swap slices, use the -M option. See “To Create a Boot Environment and Reconfigure Swap By Using a List (Command-Line Interface)” on page 427.

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

▼ To Create a Boot Environment and Reconfigure Swap By Using a List (Command-Line Interface)

If you have a long list of swap slices, create a swap list. lucreate uses this list for the swap slices in the new boot environment.

Note – The swap slice cannot be in use by any boot environment except the current boot environment or if the `-s` option is used, the source boot environment. The boot environment creation fails if the swap slice is being used by any other boot environment, whether the swap slice contains a swap, UFS, or any other file system.

---

1. **Create a list of swap slices to be used in the new boot environment. The location and name of this file is user-defined. In this example, the content of the `/etc/lu/swapslices` file is a list of devices and slices:**

   ```
   -:/dev/dsk/c0t3d0s2:swap
   -:/dev/dsk/c0t3d0s2:swap
   -:/dev/dsk/c0t4d0s2:swap
   -:/dev/dsk/c0t5d0s2:swap
   -:/dev/dsk/c1t3d0s2:swap
   -:/dev/dsk/c1t4d0s2:swap
   -:/dev/dsk/c1t5d0s2:swap
   ```

2. **Type:**

   ```
   # lucreate  [-A 'BE_description'] \
   -m mountpoint:device[,metadevice]:fs_options \
    -M slice_list  -n BE_name
   ```

   `-A` *'BE_description'*
   (Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

   `-m` *mountpoint:device[,metadevice]:fs_options* [`-m`...]
   Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

   - *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
   - *device* field can be one of the following:
     - The name of a disk device, of the form `/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z*
     - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/d*num*
     - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/*volume_name*
     - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
   - *fs_options* field can be one of the following:
     - `ufs`, which indicates a UFS file system.
     - `vxfs`, which indicates a Veritas file system.

- swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
- For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see “To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)” on page 435.

-M *slice_list*

List of -m options, which are collected in the file *slice_list*. Specify these arguments in the format that is specified for -m. Comment lines, which begin with a hash mark (#), are ignored. The -M option is useful when you have a long list of file systems for a boot environment. Note that you can combine -m and -M options. For example, you can store swap slices in *slice_list* and specify root (/) and /usr slices with -m.

The -m and -M options support the listing of multiple slices for a particular mount point. In processing these slices, lucreate skips any unavailable slices and selects the first available slice.

-n *BE_name*

The name of the boot environment to be created. *BE_name* must be unique.

EXAMPLE 35–5 Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)

In this example, swap in the new boot environment is the list of slices that are noted in the /etc/lu/swapslices file. A description, mydescription, is associated with the name second_disk.

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c02t4d0s0:ufs -m /usr:/dev/dsk/c02t4d0s1:ufs \
-M /etc/lu/swapslices -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

▼ To Create a Boot Environment and Copy a Shareable File System (Command-Line Interface)

If you want a shareable file system to be copied to the new boot environment, specify the mount point to be copied with the -m option. Otherwise, shareable file systems are shared by default, and maintain the same mount point in the vfstab file. Any updating that is applied to the shareable file system is available to both boot environments.

1. **Log in as superuser.**

2. **Create the boot environment.**

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
[-m ] mountpoint:device[,metadevice]:fs_options   -n BE_name
```

-A *'BE_description'*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

-m *mountpoint:device[,metadevice]:fs_options* [-m...]
Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to -m can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
  - The name of a disk device, of the form /dev/dsk/c*w*t*x*d*y*s*z*
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/*volume_name*
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
  - ufs, which indicates a UFS file system.
  - vxfs, which indicates a Veritas file system.
  - swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
  - For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

-n *BE_name*
The name of the boot environment to be created. *BE_name* must be unique.

EXAMPLE 35–6 Creating a Boot Environment and Copying a Shareable File System (Command-Line Interface)

In this example, the current boot environment contains two file systems, root (/) and /home. In the new boot environment, root (/) is split into two file systems, root (/) and /usr. The /home file system is copied to the new boot environment. A description, mydescription, is associated with the boot environment name second_disk.

EXAMPLE 35–6 Creating a Boot Environment and Copying a Shareable File System (Command-Line Interface) *(Continued)*

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \
-m /home:/dev/dsk/c0t4d0s4:ufs -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 36.

▼ To Create a Boot Environment From a Different Source (Command-Line Interface)

The `lucreate` command creates a boot environment that is based on the file systems in the active boot environment. If you want to create a boot environment based on a boot environment other than the active boot environment, use `lucreate` with the `-s` option.

Note – If you activate the new boot environment and need to fall back, you boot back to the boot environment that was last active, not the source boot environment.

---

1. **Log in as superuser.**
2. **Create the boot environment.**

```
# lucreate [-A 'BE_description'] -s source_BE_name
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

`-A` *'BE_description'*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

`-s` *source_BE_name*
Specifies the source boot environment for the new boot environment. The source would not be the active boot environment.

`-m` *mountpoint:device[,metadevice]:fs_options* [`-m`...]
Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:

  - The name of a disk device, of the form /dev/dsk/*cwtxdysz*
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/*volume_name*
  - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
  - `ufs`, which indicates a UFS file system.
  - `vxfs`, which indicates a Veritas file system.
  - `swap`, which indicates a swap file system. The swap mount point must be a – (hyphen).
  - For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see “To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)” on page 435.

`-n` *BE_name*

The name of the boot environment to be created. *BE_name* must be unique on the system.

EXAMPLE 35–7 Creating a Boot Environment From a Different Source (Command-Line Interface)

In this example, a boot environment is created based on the root (/) file system in the source boot environment named third_disk. Third_disk is not the active boot environment. A description, `mydescription`, is associated with the new boot environment named second_disk.

```
# lucreate -A 'mydescription' -s third_disk \
-m /:/dev/dsk/c0t4d0s0:ufs  -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 36.

▼ To Create an Empty Boot Environment for a Solaris Flash Archive (Command-Line Interface)

The `lucreate` command creates a boot environment that is based on the file systems in the active boot environment. When using the `lucreate` command with the `-s -` option, `lucreate` quickly creates an empty boot environment. The slices are reserved for the file systems specified, but no file systems are copied. The boot environment is named, but not actually created until installed with a Solaris Flash archive. When the empty boot environment is installed with an archive, file systems are installed on the reserved slices.

1. **Log in as superuser.**

2. **Create the empty boot environment.**

```
# lucreate -A 'BE_name' -s - \
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

-A *'BE_description'*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

-s -
Specifies that an empty boot environment be created.

-m *mountpoint:device[,metadevice]:fs_options* [-m...]
Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to -m can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
  - The name of a disk device, of the form /dev/dsk/c*w*t*x*d*y*s*z*
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/*volume_name*
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
  - ufs, which indicates a UFS file system.
  - vxfs, which indicates a Veritas file system.
  - swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
  - For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

-n *BE_name*
The name of the boot environment to be created. *BE_name* must be unique on the system.

EXAMPLE 35–8 Creating an Empty Boot Environment for a Solaris Flash Archive (Command-Line Interface)

In this example, a boot environment is created but contains no file systems. A description, `mydescription`, is associated with the new boot environment that is named second_disk.

```
# lucreate -A 'mydescription' -s - \
-m /:/dev/dsk/c0t1d0s0:ufs  -n second_disk
```

When creation of the empty boot environment is complete, a flash archive can be installed and can be activated (made bootable). See Chapter 36.

For an example of creating and populating an empty boot environment, see "Example of Creating an Empty Boot Environment and Installing a Solaris Flash Archive (Command-Line Interface)" on page 495.

Figure 35–4 shows the creation of an empty boot environment.

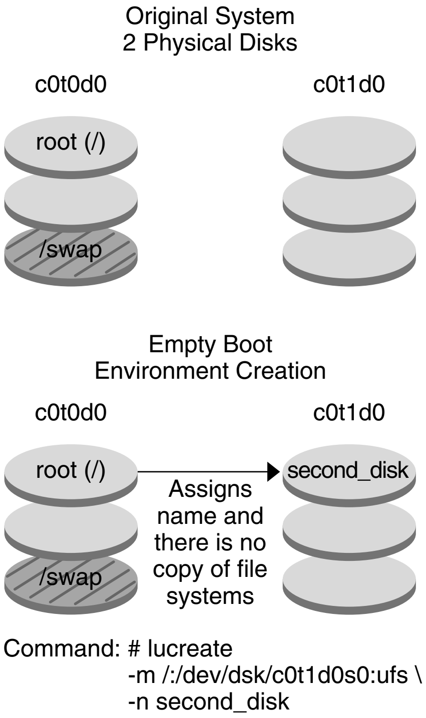


FIGURE 35–4 Creating an Empty Boot Environment

▼ To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)

When you create a boot environment, Solaris Live Upgrade uses Solaris Volume Manager technology to create RAID-1 volumes. When creating a boot environment, you can use Solaris Live Upgrade to manage the following tasks.

- Remove a single-slice concatenation (submirror) from a RAID-1 volume (mirror). The contents can be saved to become the content of the new boot environment if necessary. Because the contents are not copied, the new boot environment can be quickly created. After the submirror is detached from a mirror, it is no longer part of the original mirror. Reads and writes to the submirror are no longer performed through the mirror.
- Create a boot environment that contains a mirror.
- Attach a single-slice concatenation to the newly created mirror.

To use the mirroring capabilities of Solaris Live Upgrade, you must create a state database and a state database replica. A state database stores information on disk about the state of your Solaris Volume Manager configuration.

- For information about creating a state database, see "State Database (Overview)" in *Solaris Volume Manager Administration Guide*.
- For an overview of Solaris Volume Manager and the tasks that Solaris Live Upgrade can provide, see "Creating a Boot Environment With Mirrored File Systems" on page 382.
- For in-depth information about complex Solaris Volume Manager configurations that are not allowed when using Solaris Live Upgrade, see "Storage Management Concepts" in *Solaris Volume Manager Administration Guide*.

1. **Log in as superuser.**
2. **To create the new boot environment, type:**

```
# lucreate [-A 'BE_description']  \
-m mountpoint:device[,metadevice]:fs_options [-m...] \
-n BE_name
```

`-A` *'BE_description'*
(Optional) Enables the creation of a boot environment description that is associated with the boot environment name *BE_name*. The description can be any length and can contain any characters.

`-m` *mountpoint:device[,metadevice]:fs_options* [`-m`...]
Specifies the file systems' configuration of the new boot environment in the `vfstab`. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.

- *device* field can be one of the following:
  - The name of a disk device, of the form /dev/dsk/c*w*t*x*d*y*s*z*
  - The name of a Solaris Volume Manager volume, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/md/vxfs/dsk/d*num*
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following types of file systems and keywords:
  - ufs, which indicates a UFS file system.
  - vxfs, which indicates a Veritas file system.
  - swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
  - For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device.
    - mirror creates a RAID–1 volume or mirror on the specified device. In subsequent -m options, you must specify attach to attach at least one concatenation to the new mirror. The specified device must be correctly named. For example, a logical device name of /dev/md/dsk/d10 or the short name of d10 can serve as a mirror name. For more information about naming devices, see "Overview of Solaris Volume Manager Components" in *Solaris Volume Manager Administration Guide*.
    - detach removes a concatenation from a volume that is associated with a specified mount point. The volume does not need to be specified.
    - attach attaches a concatenation to the mirror that is associated with a specified mount point. The physical disk slice that is specified is made into a single device concatenation for attaching to the mirror. To specify a concatenation to attach to a disk, you append a comma and the name of that concatenation to the device name. If you omit the comma and the concatenation name, lucreate selects a free volume for the concatenation.

      lucreate allows you to create only concatenations that contain a single physical slice. This command allows you to attach up to three concatenations to a mirror.
    - preserve saves the existing file system and its contents. This keyword enables you to bypass the copying process that copies the contents of the source boot environment. Saving the contents enables a quick creation of the new boot environment. For a particular mount

point, you can use preserve with only one physical device. When you use preserve, lucreate checks that the device's contents are suitable for a specified file system. This check is limited and cannot guarantee suitability.

The preserve keyword can be used with both a physical slice and Solaris Volume Manager volume.

- If you use the preserve keyword when the UFS file system is on a physical slice, the contents of the UFS file system is saved on the slice. In the following example of the -m option, the preserve keyword saves the contents of the physical device c0t0d0s0 as the file system for the mount point for the root (/) file system.

  ```
  -m /:c0t0d0s0:preserve,ufs
  ```

- If you use the preserve keyword when the UFS file system is on a volume, the contents of the UFS file system is saved on the volume.

  In the following example of the -m option, the preserve keyword saves the contents of the RAID-1 volume (mirror) d10 as the file system for the mount point for the root (/) file system.

  ```
  -m /:d10:preserve,ufs
  ```

  In the following example of the -m option, a RAID-1 volume (mirror) d10 is configured as the file system for the mount point for the root (/) file system. The single-slice concatenation d20 is detached from its current mirror. d20 is attached to mirror d10. The root (/) file system is preserved on submirror d20.

  ```
  -m /:d10:mirror,ufs -m /:d20:detach,attach,preserve
  ```

-n *BE_name*

The name of the boot environment to be created. *BE_name* must be unique on the system.

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

EXAMPLE 35–9 Creating a Boot Environment With a Mirror and Specifying Devices (Command Line)

In this example, the mount points for the file systems are specified by using the -m option.

- A description, mydescription, is associated with the name another_disk.
- lucreate configures a UFS file system for the mount point root (/). A mirror, d10, is created. This mirror is the receptacle for the current boot environment's root (/) file system that is copied to the mirror d10. All data on the mirror d10 is overwritten.

EXAMPLE 35–9 Creating a Boot Environment With a Mirror and Specifying Devices (Command Line) *(Continued)*

- Two slices, c0t0d0s0 and c0t1d0s0, are submirrors, d1 and d2. These two submirrors are added to mirror d10.
- The new boot environment is named another_disk.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,d1:attach \
-m /:/dev/dsk/c0t1c0s0,d2:attach -n another_disk
```

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

EXAMPLE 35–10 Creating a Boot Environment With a Mirror and Not Specifying a Submirror Name (Command Line)

In this example, the mount points for the file systems are specified by using the -m option.

- A description, mydescription, is associated with the name another_disk.
- lucreate configures a UFS file system for the mount point root (/). A mirror, d10, is created. This mirror is the receptacle for the current boot environment's root (/) file system that is copied to the mirror d10. All data on the mirror d10 is overwritten.
- Two slices, c0t0d0s0 and c0t1d0s0, are specified to be used as submirrors. The submirrors are not specified, but the lucreate command chooses names from a list of available volume names. These two submirrors are attached to mirror d10.
- The new boot environment is named another_disk.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0:attach \
-m /:/dev/dsk/c0t1d0s0:attach -n another_disk
```

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

EXAMPLE 35–11 Creating a Boot Environment and Detaching a Submirror (Command Line)

In this example, the mount points for the file systems are specified by using the -m option.

- A description, mydescription, is associated with the name another_disk.
- lucreate configures a UFS file system for the mount point root (/). A mirror, d10, is created.

EXAMPLE 35–11 Creating a Boot Environment and Detaching a Submirror (Command Line) *(Continued)*

- Slice c0t0d0s0 is removed from its current mirror. The slice is specified to be submirror d1 and is added to mirror d10. The contents of the submirror, the root (/) file system, are saved and no copy occurs. Slice c0t1d0s0 is submirror d2 and is added to mirror d10.
- The new boot environment is named another_disk.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,d1:detach,attach,preserve \
-m /:/dev/dsk/c0t1d0s0,d2:attach -n another_disk
```

This example can be abbreviated as in the following example. The physical and logical device names are shortened. The specifiers for the submirrors d1 and d2 are omitted.

```
# lucreate -A 'mydescription' \
-m /:d10:ufs,mirror \
-m /:c0t0d0s0:detach,attach,preserve \
-m /:c0t1d0s0:attach -n another_disk
```

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

EXAMPLE 35–12 Creating a Boot Environment, Detaching a Submirror, and Saving Its Contents (Command Line)

In this example, the mount points for the file systems are specified by using the -m option.

- A description, mydescription, is associated with the name another_disk.
- lucreate configures a UFS file system for the mount point root (/). A mirror, d20, is created.
- Slice c0t0d0s0 is removed from its current mirror and added to the mirror d20. The name of the submirror is not specified. The contents of the submirror, the root (/) file system, are saved and no copy occurs.
- The new boot environment is named another_disk.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d20:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0:detach,attach,preserve \
-n another_disk
```

When the creation of the new boot environment is complete, the boot environment can be upgraded and can be activated (made bootable). See Chapter 36.

EXAMPLE 35–13 Creating a Boot Environment With Two Mirrors (Command Line)

In this example, the mount points for the file systems are specified by using the `-m` option.

- A description, `mydescription`, is associated with the name `another_disk`.
- `lucreate` configures a UFS file system for the mount point root (/). A mirror, `d10`, is created. This mirror is the receptacle for the current boot environment's root (/) file system that is copied to the mirror `d10`. All data on the mirror `d10` is overwritten.
- Two slices, `c0t0d0s0` and `c0t1d0s0`, are submirrors, `d1` and `d2`. These two submirrors are added to mirror `d10`.
- `lucreate` configures UFS file system for the mount point `/opt`. A mirror, `d11`, is created. This mirror is the receptacle for the current boot environment's `/opt` file system that is copied to the mirror `d11`. All data on the mirror `d11` is overwritten.
- Two slices, `c2t0d0s1` and `c3t1d0s1`, are submirrors, `d3` and `d4`. These two submirrors are added to mirror `d11`.
- The new boot environment is named `another_disk`.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,d1:attach \
-m /:/dev/dsk/c0t1d0s0,d2:attach \
-m /opt:/dev/md/dsk/d11:ufs,mirror \
-m /opt:/dev/dsk/c2t0d0s1,d3:attach \
-m /opt:/dev/dsk/c3t1d0s1,d4:attach -n another_disk
```

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

▼ To Create a Boot Environment and Customize the Content (Command-Line Interface)

The content of the file system on the new boot environment can be modified by using the following options. Directories and files are not copied to the new boot environment.

1. **Log in as superuser.**
2. **To create the new boot environment, type:**

```
# lucreate -m mountpoint:device[,metadevice]:fs_options  [-m ...]   \
[-x exclude_dir]  [-y include]  [-Y include_list_file]  [-f exclude_list_file]\
 [-z filter_list]  [-I]  -n BE_name
```

-m *mountpoint:device[,metadevice]:fs_options* [-m ...]

Specifies the file systems' configuration of the new boot environment in the vfstab. The file systems that are specified as arguments to -m can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
  - The name of a disk device, of the form /dev/dsk/c*w*t*x*d*y*s*z*
  - The name of a Solaris Volume Manager volume, of the form /dev/md/dsk/d*num*
  - The name of a Veritas Volume Manager volume, of the form /dev/md/vxfs/dsk/d*num*
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
  - ufs, which indicates a UFS file system.
  - vxfs, which indicates a Veritas file system.
  - swap, which indicates a swap file system. The swap mount point must be a – (hyphen).
  - For file systems that are logical devices (mirrors), several keywords specify actions to be taken on the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

-x *exclude_dir*

Excludes files and directories by not copying them to the new boot environment. You can use multiple instances of this option to exclude more than one file or directory.

exclude_dir is the name of the directory or file.

-y *include_dir*

Copies directories and files that are listed to the new boot environment. This option is used when you have excluded a directory, but want to restore individual subdirectories or files.

include_dir is the name of the subdirectory or file to be included.

-Y *list_filename*

Copies directories and files from a list to the new boot environment. This option is used when you have excluded a directory, but want to restore individual subdirectories or files.

- *list_filename* is the full path to a file that contains a list.
- The *list_filename* file must contain one file per line.

- If a line item is a directory, all subdirectories and files beneath that directory are included. If a line item is a file, only that file is included.

`-f` *list_filename*
Uses a list to exclude directories and files by not copying them to the new boot environment.

- *list_filename* is the full path to a file that contains a list.
- The *list_filename* file must contain one file per line.

`-z` *list_filename*
Uses a list to copy directories and files to the new boot environment. Each file or directory in the list is noted with a plus "+" or minus "-". A plus indicates an included file or directory and the minus indicates an excluded file or directory.

- *list_filename* is the full path to a file that contains a list.
- The *list_filename* file must contain one file per line. A space must follow the plus or minus before the file name.
- If a line item is a directory and is indicated with a + (plus), all subdirectories and files beneath that directory are included. If a line item is a file and is indicated with a + (plus), only that file is included.

`-I`
Overrides the integrity check of system files. Use this option with caution.

To prevent you from removing important system files from an boot environment, `lucreate` runs an integrity check. This check examines all files that are registered in the system package database and stops the boot environment creation if any files are excluded. Use of this option overrides this integrity check. This option creates the boot environment more quickly, but might not detect problems.

`-n` *BE_name*
The name of the boot environment to be created. *BE_name* must be unique on the system.

EXAMPLE 35–14 Creating a Boot Environment and Excluding Files (Command Line)

In this example, the new boot environment is named `second_disk`. The source boot environment contains one file system, root (`/`). In the new boot environment, the `/var` file system is split from root (`/`) and put on another slice. The `lucreate` command configures a UFS file system for the mount points root (`/`) and `/var`. Also, two `/var` mail files, `root` and `staff` are not copied to the new boot environment. Swap is automatically shared between the source and the new boot environment.

```
# lucreate -n second_disk \
-m /:/dev/dsk/c0t1d0s0:ufs -m /var/mail:c0t2d0s0:ufs  \
 -x /var/mail/root -x /var/mail/staff
```

EXAMPLE 35–15 Creating a Boot Environment and Excluding and Including Files (Command Line)

In this example, the new boot environment is named `second_disk`. The source boot environment contains one file system for the operating environment, root (/). The source also contains a file system that is named `/mystuff`. `lucreate` configures a UFS file system for the mount points root (/) and `/mystuff`. Only two directories in `/mystuff` are copied to the new boot environment, `/latest` and `/backup`. Swap is automatically shared between the source and the new boot environment.

```
# lucreate -n second_disk \
-m /:/dev/dsk/c01t0d0s0:ufs -m /mystuff:c1t1d0s0:ufs  \
 -x /mystuff -y /mystuff/latest -y /mystuff/backup
```

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 36.

CHAPTER 36

Upgrading With Solaris Live Upgrade (Tasks)

This chapter explains how to use Solaris Live Upgrade to upgrade and activate an inactive boot environment. It also explains how to recover from an activation failure. This chapter contains the following sections:

You can use Solaris Live Upgrade with menus or by using the command-line interface (CLI). Procedures are documented for both interfaces. These procedures do not exhaust the possibilities for using Solaris Live Upgrade. For more information about commands, see Chapter 39 and the appropriate man pages, which more fully document CLI options.

Task Map: Upgrading a Boot Environment

TABLE 36–1 Task Map: Upgrading With Solaris Live Upgrade

Task	Description	For Instructions
Either upgrade a boot environment or install a Solaris Flash archive	■ Upgrade the inactive boot environment with an operating environment image ■ Install a Solaris Flash archive on an inactive boot environment	■ "Upgrading a Boot Environment" on page 446 ■ "Installing Solaris Flash Archives on a Boot Environment" on page 456
Activate an inactive boot environment	Makes changes effective and switches the inactive boot environment to active	"Activating a Boot Environment" on page 461
(optional) Switch back if a failure occurs when activating	Reactivates to the original boot environment if a failure occurs	"Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)" on page 465

Upgrading a Boot Environment

Use the Upgrade menu or `luupgrade` command to upgrade a boot environment. This section provides the procedure for upgrading an inactive boot environment from files that are located on the following media:

- NFS server
- Local file
- Local tape
- Local device, including DVD or CD

Note – If the installation requires more than one CD, you must use the command-line interface procedure for upgrading. See "To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)" on page 449.

Guidelines for Upgrading

When you upgrade a boot environment with the latest operating environment, you do not affect the active boot environment. The new files merge with the inactive boot environment critical file systems, but shareable file systems are not changed.

Rather than upgrading, if you have created a Solaris Flash archive, you could install the archive on an inactive boot environment. The new files overwrite the inactive boot environment critical file systems, but shareable file systems are not changed. See "Installing Solaris Flash Archives on a Boot Environment" on page 456.

You can upgrade an inactive boot environment that contains any combination of physical disk slices, Solaris Volume Manager volumes, or Veritas Volume Manager volumes. The slice that is chosen for the root (`/`) file system must be a single-slice concatenation that is included in a RAID–1 volume (mirror). For procedures about creating a boot environment with mirrored file systems, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

Adding and Removing Packages With Solaris Live Upgrade

Caution – When upgrading and adding and removing packages or patches, Solaris Live Upgrade requires packages or patches that comply with the SVR4 advanced packaging guidelines. While Sun packages conform to these guidelines, Sun cannot guarantee the conformance of packages from third-party vendors. If a package violates these guidelines, the package can cause the package-addition software to fail or can alter the active boot environment.

For more information on adding and removing packages with Solaris Live Upgrade, see the following:

- For planning information, see "Managing Packages and Patches With Solaris Live Upgrade" on page 398.
- For information on the command and options, see the man page, `luupgrade`(1M).
- For more information on packaging requirements, see Appendix G.

▼ To Upgrade an Operating System Image on a Boot Environment (Character Interface)

To upgrade by using this procedure, you must use only one CD, a DVD, or a combined installation image. For an installation that requires more than one CD, you must use the procedure: "To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)" on page 449.

1. **From the Solaris Live Upgrade main menu, select Upgrade.**

   The Upgrade menu screen is displayed.

2. **Type the new boot environment's name.**

3. **Type the path to where the Solaris installation image is located.**

Installation Media Type	Description
Network File System	Specify the path to the network file system where the installation image is located.
Local file	Specify the path to the local file system where the installation image is located.
Local tape	Specify the local tape device and the position on the tape where the installation image is located.
Local device, DVD, or CD	Specify the local device and the path to the installation image.

   - If you are using DVD or one CD, type the path to that disc, as in this example:

     ```
     Package Media: /cdrom/Solaris_9/s0
     ```

   - If you have a combined image on the network, type the path to the network file system as in this example:

     ```
     Package Media: /net/installmachine/export/Solaris_9/os_image
     ```

4. **To upgrade, press F3.**

5. **When the upgrade is completed, the main menu is displayed.**

▼ To Upgrade an Operating System Image on a Boot Environment (Command-Line Interface)

To upgrade by using this procedure, you must use only one CD, a DVD, or a combined installation image. If the installation requires more than one CD, you must use the procedure: "To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)" on page 449.

1. **Log in as superuser.**
2. **Indicate the boot environment to upgrade and the path to the installation software by typing:**

```
# luupgrade -u -n BE_name -s os_image_path
```

-u Upgrades an operating system image on a boot environment

-n *BE_name* Specifies the name of the boot environment that is to be upgraded

-s *os_image_path* Specifies the path name of a directory that contains an operating system image

EXAMPLE 36–1 Upgrading an Operating System Image on a Boot Environment (Command-Line Interface)

In this example, the second_disk boot environment is upgraded.

```
# luupgrade -u -n second_disk \
-s /net/installmachine/export/Solaris_9/OS_image
```

▼ To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)

If the operating system image resides on two or more CDs, use this upgrade procedure.

1. **Log in as superuser.**
2. **Indicate the boot environment to upgrade and the path to the installation software by typing:**

```
# luupgrade -u -n BE_name -s os_image_path
```

-u Upgrades an operating system image on a boot environment

-n *BE_name* Specifies the name of the boot environment that is to be upgraded

-s *os_image_path* Specifies the path name of a directory that contains an operating system image

3. **When the installer is finished with the contents of the first CD, insert the second CD.**
4. **This step is identical to the previous step, but the `-u` option is replaced by the `-i` option. Also, choose to run the installer on the second CD with menus or with text.**

- This command runs the installer on the second CD with menus.

  ```
  # luupgrade -i -n BE_name -s os_image_path
  ```

- This command runs the installer on the second CD with text and requires no user interaction.

  ```
  # luupgrade -i -n BE_name -s os_image_path -O '-nodisplay -noconsole'
  ```

`-i`	Looks for an installation program on the specified medium and runs that program. The installer program is specified with `-s`.
`-n` *BE_name*	Specifies the name of the boot environment that is to be upgraded.
`-s` *os_image_path*	Specifies the path name of a directory that contains an operating system image.
`-O '-nodisplay -noconsole'`	(Optional) Runs the installer on the second CD in text mode and requires no user interaction.

The boot environment is ready to be activated. See "Activating a Boot Environment" on page 461.

EXAMPLE 36–2 SPARC: Upgrading an Operating System Image From Multiple CDs (Command-Line Interface)

In this example, the second_disk boot environment is upgraded and the installation image is on two CDs: the Solaris Software 1 of 2 and the Solaris Software 2 of 2 CDs. The `-u` option determines if sufficient space for all the packages is on the CD set. The `-O` option with the `-nodisplay` and `-noconsole` options prevents the character user interface from displaying after the reading of the second CD. If you use these options, you are not prompted to input information. Omit these options to display the interface.

Insert the Solaris Software 1 of 2 CD and type:

```
# luupgrade -u -n second_disk -s /dev/cdrom/cdrom0/s0
```

Insert the Solaris Software 2 of 2 CD and type:

```
# luupgrade -i -n second_disk -s /dev/cdrom/cdrom0 -O '-nodisplay \
-noconsole'
```

Note – Use the `luupgrade` command with the `-i` option to install any additional CDs.

▼ To Create a Profile to be Used by Solaris Live Upgrade

This procedure shows you how to create a profile for use with Solaris Live Upgrade. You can use this profile to upgrade an inactive boot environment by using the `luupgrade` command with the `-j` option.

For procedures to use this profile, see the following sections:

- For an upgrade with a profile, see "To Upgrade With a Profile by Using Solaris Live Upgrade (Command-Line Interface)" on page 455.
- For a Solaris Flash installation with a profile, see "To Install a Solaris Flash Archive With a Profile (Command-Line Interface)" on page 459.

1. **Use a text editor to create a text file.**

   Name the file descriptively. Ensure that the name of the profile reflects how you intend to use the profile to install the Solaris software on a system. For example, you might name this profile `upgrade_Solaris_9`.

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

   Only the upgrade keywords in the following tables can be used in a Solaris Live Upgrade profile.

   - For an installation using the `Install_type` keyword values of `upgrade` or `flash_install`, see Table 36–2.
   - For an installation using the `Install_type` keyword value of `flash_update`, see Table 36–3.

TABLE 36–2 Solaris Live Upgrade Profile Keywords and Descriptions

Keyword	Description
(Required) `Install_type`	Defines whether to upgrade the existing Solaris environment on a system or install a Solaris Flash archive on the system. Use the following values with this keyword: ■ `upgrade` for an upgrade ■ `flash_install` for a Solaris Flash installation ■ `flash_update` for a Solaris Flash differential installation
(Required for a Solaris Flash archive) `archive_location`	Retrieves a Solaris Flash archive from a designated location. For a list of values that can be used with this keyword, see "`archive_location` Keyword" on page 342.
(Optional) `cluster` (adding or deleting clusters)	Designates whether a cluster is to be added or deleted from the software group that is to be installed on the system. For a list of values that can be used with this keyword, see "`cluster` Profile Keyword (Adding Software Groups)" on page 352.

TABLE 36–2 Solaris Live Upgrade Profile Keywords and Descriptions *(Continued)*

Keyword	Description
(Optional) geo	Designates the regional locale or locales that you want to install on a system or to add when upgrading a system. For a list of values that can be used with this keyword, see "geo Profile Keyword" on page 359.
(Optional) isa_bits	Specifies whether 64-bit or 32-bit Solaris 9 packages are to be installed. For a list of values that can be used with this keyword, see "isa_bits Profile Keyword" on page 361.
(Optional) local_customization	Before you install a Solaris Flash archive on a clone system, you can create custom scripts to preserve local configurations on the clone system. The local_customization keyword designates the directory where you have stored these scripts. The value is the path to the script on the clone system. For information on predeployment and postdeployment scripts, see "Creating Customization Scripts" on page 221.
(Optional) locale	Designates the locale packages you want to install or add when upgrading. For a list of values that can be used with this keyword, see "locale Profile Keyword" on page 363.
(Optional) package	Designates whether a package is to be added to or deleted from the software group that is to be installed on the system. For a list of values that can be used with this keyword, see "package Profile Keyword" on page 365.

TABLE 36–3 Solaris Live Upgrade Profile Keywords and Descriptions For Creating a Differential Archive

Keyword	Description
(Required) Install_type	Defines the installation to install a Solaris Flash archive on the system. The value for a differential archive is flash_update.
(Required) archive_location	Retrieves a Solaris Flash archive from a designated location. For a list of values that can be used with this keyword, see "archive_location Keyword" on page 342.
(Optional) forced_deployment	Forces the installation of a Solaris Flash differential archive onto a clone system that is different than the software expects. If you use forced_deployment, all new files are deleted to bring the clone system to the expected state. If you are not certain that you want files to be deleted, use the default, which protects new files by stopping the installation.

TABLE 36–3 Solaris Live Upgrade Profile Keywords and Descriptions For Creating a Differential Archive *(Continued)*

Keyword	Description
(Optional) `local_customization`	Before you install a Solaris Flash archive on a clone system, you can create custom scripts to preserve local configurations on the clone system. The `local_customization` keyword designates the directory where you have stored these scripts. The value is the path to the script on the clone system. For information about predeployment and postdeployment scripts, see "Creating Customization Scripts" on page 221.
(Optional) `no_content_check`	When installing a clone system with a Solaris Flash differential archive, you can use the `no_content_check` keyword to ignore file-by-file validation. File-by-file validation ensures that the clone system is a duplicate of the master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.
(Optional) `no_master_check`	When installing a clone system with a Solaris Flash differential archive, you can use the `no_master_check` keyword to ignore a check of files. Clone system files are not checked. A check would ensure the clone was built from the original master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.

3. **Save the profile in a directory on the local system.**
4. **Ensure that `root` owns the profile and that the permissions are set to 644.**
5. **Test the profile (optional).**

   For a procedure to test the profile, see "To Test a Profile to Be Used by Solaris Live Upgrade" on page 454.

EXAMPLE 36–3 Creating a Solaris Live Upgrade Profile

In this example, a profile provides the upgrade parameters. This profile is to be used to upgrade an inactive boot environment with the Solaris Live Upgrade `luupgrade` command and the `-u` and `-j` options. This profile adds a package and a cluster. A regional locale and additional locales are also added to the profile. If you add locales to the profile, make sure that you have created a boot environment with additional disk space.

```
# profile keywords         profile values
# ----------------         -------------------
  install_type             upgrade
  package                  SUNWxwman add
  cluster                  SUNWCacc add
  geo                      C_Europe
  locale                   zh_TW
  locale                   zh_TW.BIG5
```

EXAMPLE 36–3 Creating a Solaris Live Upgrade Profile *(Continued)*

```
locale                    zh_TW.UTF-8
locale                    zh_HK.UTF-8
locale                    zh_HK.BIG5HK
locale                    zh
locale                    zh_CN.GB18030
locale                    zh_CN.GBK
locale                    zh_CN.UTF-8
```

EXAMPLE 36–4 Creating a Solaris Live Upgrade Profile to Install a Differential Archive

The following example of a profile is to be used by Solaris Live Upgrade to install a differential archive on a clone system. Only files that are specified by the differential archive are added, deleted, or changed. The Solaris Flash archive is retrieved from an NFS server. Because the image was built by the original master system, the clone system is not checked for a valid system image. This profile is to be used with the Solaris Live Upgrade `luupgrade` command and the `-u` and `-j` options.

```
# profile keywords         profile values
# ----------------         -------------------
 install_type              flash_update
 archive_location          nfs installserver:/export/solaris/flasharchive/solarisarchive
 no_master_check
```

To use the `luupgrade` command to install the differential archive, see "To Install a Solaris Flash Archive With a Profile (Command-Line Interface)" on page 459.

To Test a Profile to Be Used by Solaris Live Upgrade

After you create a profile, use the `luupgrade` command to test the profile. By looking at the installation output that is generated by `luupgrade`, you can quickly determine if a profile works as you intended.

- **Test the profile.**

  ```
  # luupgrade -u -n BE_name -D -s os_image_path -j profile_path
  ```

  `-u` Upgrades an operating system image on a boot environment.

  `-n` *BE_name* Specifies the name of the boot environment that is to be upgraded.

  `-D` `luupgrade` command uses the selected boot environment's disk configuration to test the profile options passed with the `-j` option.

`-s` *os_image_path* Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.

`-j` *profile_path* Path to a profile that is configured for an upgrade. The profile must be in a directory on the local machine.

EXAMPLE 36–5 Testing a Profile by Using Solaris Live Upgrade

In the following example, the profile is named `upgrade_Solaris_9`. The profile is successfully tested on the inactive boot environment named, `u1b08`.

```
# luupgrade -u -n u1b08 -D -s /net/installsvr/export/u1/combined.u1wos \
 -j /var/tmp/flash_profile
Validating the contents of the media /net/installsvr/export/u1/combined.u1wos.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains Solaris version 9.
Locating upgrade profile template to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live Upgrade requests.
Creating upgrade profile for BE u1b08.
Determining packages to install or upgrade for BE u1b08.
Simulating the operating system upgrade of the BE u1b08.
The operating system upgrade simulation is complete.
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of the
upgrade operation.
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of
cleanup operations required.
The Solaris upgrade of the boot environment u1b08 is complete.
```

You can now use the profile to upgrade an inactive boot environment.

▼ To Upgrade With a Profile by Using Solaris Live Upgrade (Command-Line Interface)

This procedure provides step-by-step instructions for upgrading an operating environment by using a profile.

If you want to install a Solaris Flash archive by using a profile, see "To Install a Solaris Flash Archive With a Profile (Command-Line Interface)" on page 459.

If you added locales to the profile, make sure that you have created a boot environment with additional disk space.

1. **Log in as superuser.**

2. **Create a profile.**

   See Table 36–2 for a list of upgrade keywords that can be used in a Solaris Live Upgrade profile.

3. **Type:**

   ```
   # luupgrade -u -n BE_name -s os_image_path -j profile_path
   ```

`-u`	Upgrades an operating system image on a boot environment
`-n` *BE_name*	Specifies the name of the boot environment that is to be upgraded.
`-s` *os_image_path*	Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.
`-j` *profile_path*	Path to a profile. The profile must be in a directory on the local machine. For information on creating a profile, see "To Create a Profile to be Used by Solaris Live Upgrade" on page 451.

   The boot environment is ready to be activated.

EXAMPLE 36–6 Upgrading a Boot Environment By Using a Custom JumpStart Profile (Command-Line Interface)

In this example, an archive is installed on the `second_disk` boot environment. The `-j` option is used to access the profile. The boot environment is then ready to be activated. To create a profile, see "To Create a Profile to be Used by Solaris Live Upgrade" on page 451.

```
# luupgrade -u -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

Installing Solaris Flash Archives on a Boot Environment

This section provides the procedure for using Solaris Live Upgrade to install Solaris Flash archives that are stored on the following:

- HTTP server
- FTP server - Use this path from the command line only
- NFS server

- Local file
- Local tape
- Local device, including DVD or CD

Note – For examples of the correct syntax for paths that are associated with archive storage, see “`archive_location` Keyword” on page 342.

Installing a Solaris Flash archive overwrites all files on the new boot environment except for shared files.

To use the Solaris Flash installation feature, you need to have installed the master system and to have created the Solaris Flash archive. For more information about Solaris Flash, see Chapter 19. To create a Solaris Flash archive, see the following procedures:

- “To Create a Solaris Flash Archive for an Initial Installation” on page 226
- “To Create a Solaris Flash Differential Archive With an Updated Master Image” on page 231
- “To Create a Solaris Flash Differential Archive By Using Solaris Live Upgrade” on page 233

To Install a Solaris Flash Archive on a Boot Environment (Character Interface)

1. **From the Solaris Live Upgrade main menu, select Flash.**

   The Flash an Inactive Boot Environment menu is displayed.

2. **Type the name of the boot environment where you want to install the Solaris Flash archive and the location of the installation media:**

   ```
   Name of Boot Environment: Solaris_9
   Package media: /net/install-svr/export/Solaris_9/latest
   ```

3. **Press F1 to add an archive.**

   An Archive Selection submenu is displayed.

   ```
   Location            - Retrieval Method
   <No Archives added> - Select ADD to add archives
   ```

   This menu enables you to build a list of archives. To add or remove archives, proceed with the following steps.

   a. **To add an archive to the menu, press F1.**

      A Select Retrieval Method submenu is displayed.

```
HTTP
NFS
Local File
Local Tape
Local Device
```

b. **On the Select Retrieval Method menu, select the location of the Solaris Flash archive.**

Media Selected	Prompt
HTTP	Specify the URL and proxy information that is needed to access the Solaris Flash archive.
NFS	Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.
Local file	Specify the path to the local file system where the Solaris Flash archive is located.
Local tape	Specify the local tape device and the position on the tape where the Solaris Flash archive is located.
Local device	Specify the local device, the path to the Solaris Flash archive, and the type of file system on which the Solaris Flash archive is located.

A Retrieval submenu is displayed, similar to the following example, which depends on the media you selected.

```
NFS Location:
```

c. **Type the path to the archive, as in the following example.**

```
NFS Location: host:/path/to archive.flar
```

d. **Press F3 to add the archive to the list.**

e. **(Optional) To remove an archive from the menu, press F2.**

f. **When the list contains the archives that you want to install, press F6 to exit.**

4. **Press F3 to install one or more archives.**

   The Solaris Flash archive is installed on the boot environment. All files on the boot environment are overwritten, except for shareable files.

   The boot environment is ready for activation. See "To Activate a Boot Environment (Character Interface)" on page 463.

▼ To Install a Solaris Flash Archive on a Boot Environment (Command-Line Interface)

1. **Log in as superuser.**

2. **Type:**

   ```
   # luupgrade -f -n BE_name -s os_image_path -a archive
   ```

   `-f` Indicates to install an operating system from a Solaris Flash archive.

   `-n` *BE_name* Specifies the name of the boot environment that is to be installed with an archive.

   `-s` *os_image_path* Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.

   `-a` *archive* Path to the Solaris Flash archive when the archive is available on the local file system. The operating system image versions specified with the `-s` option and the `-a` option must be identical.

EXAMPLE 36–7 Installing Solaris Flash Archives on a Boot Environment (Command-Line Interface)

In this example, an archive is installed on the `second_disk` boot environment. The archive is located on the local system. The operating systems versions for the `-s` and `-a` options are both Solaris 9 releases. All files are overwritten on `second_disk` except shareable files.

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_9/OS_image \
-a /net/server/archive/Solaris_9
```

The boot environment is ready to be activated.

▼ To Install a Solaris Flash Archive With a Profile (Command-Line Interface)

This procedure provides the steps to install a Solaris Flash archive or differential archive by using a profile.

If you added locales to the profile, make sure that you have created a boot environment with additional disk space.

1. **Log in as superuser.**

2. **Create a profile.**

   See Table 36–2 for a list of keywords that can be used in a Solaris Live Upgrade profile.

3. **Type:**

   ```
   # luupgrade -f -n BE_name -s os_image_path -j profile_path
   ```

   `-f` Indicates to install an operating system from a Solaris Flash archive.

   `-n` *BE_name* Specifies the name of the boot environment that is to be upgraded.

   `-s` *os_image_path* Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.

   `-j` *profile_path* Path to a profile that is configured for a flash installation. The profile must be in a directory on the local machine.

   The boot environment is ready to be activated.

EXAMPLE 36–8 Install a Solaris Flash archive on a Boot Environment With a Profile (Command-Line Interface)

In this example, a profile provides the location of the archive to be installed.

```
# profile keywords         profile values
# ----------------         -------------------
 install_type              flash_install
 archive_location          nfs installserver:/export/solaris/flasharchive/solarisarchive
```

After creating the profile, you can run the `luupgrade` command and install the archive. The `-j` option is used to access the profile.

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

The boot environment is then ready to be activated. To create a profile, see “To Create a Profile to be Used by Solaris Live Upgrade” on page 451.

▼ To Install a Solaris Flash Archive With a Profile Keyword (Command-Line Interface)

This procedure enables you to install a Solaris Flash archive and use the `archive_location` keyword at the command line rather than from a profile file. You can quickly retrieve an archive without the use of a profile file.

1. **Log in as superuser.**

2. **Type:**

   ```
   # luupgrade -f -n BE_name -s os_image_path -J 'profile'
   ```

   `-f` Indicates to upgrade an operating system from a Solaris Flash archive.

   `-n` *BE_name* Specifies the name of the boot environment that is to be upgraded.

   `-s` *os_image_path* Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.

   `-J` *'profile'* Specifies the `archive_location` profile keyword. For the keyword values, see "`archive_location` Keyword" on page 342.

   The boot environment is ready to be activated.

EXAMPLE 36–9 Installing a Solaris Flash Archive By Using a Profile Keyword (Command-Line Interface)

In this example, an archive is installed on the `second_disk` boot environment. The `-J` option and the `archive_location` keyword is used to retrieve the archive. All files are overwritten on `second_disk` except shareable files.

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-J 'archive_location http://example.com/myflash.flar'
```

Activating a Boot Environment

Activating a boot environment makes it bootable on the next reboot of the system. You can also switch back quickly to the original boot environment if a failure occurs on booting the newly active boot environment. See "Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)" on page 465.

To successfully activate a boot environment, that boot environment must meet the following conditions:

- The boot environment must have a status of "complete." To check status, see "Displaying the Status of All Boot Environments" on page 472.
- If the boot environment is not the current boot environment, you cannot have mounted the partitions of that boot environment by using `lumount`(1M) or `mount`(1M).
- The boot environment that you want to activate cannot be involved in a comparison operation. See "Comparing Boot Environments" on page 477.

Note – If you want to reconfigure swap, make this change prior to booting the inactive boot environment. By default, all boot environments share the same swap devices. To reconfigure swap, see "To Create a New Boot Environment," Step 9 or "To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)" on page 426.

▼ x86: (Optional) To Update the Boot Diskette Before Activating

If you're not using the Solaris Device Configuration Assistant *x86 Platform Edition* diskette to boot the system, you can skip this procedure. If you use the Device Configuration Assistant to boot, you need to update the boot diskette. This procedure matches a boot diskette to your release by overwriting the existing diskette or writing to a new diskette.

1. **Insert the Solaris Device Configuration Assistant *x86 Platform Edition* diskette. This is either the existing diskette that is overwritten or a new diskette.**

2. **Update the boot diskette with the latest image for this release.**

3. **Eject the boot diskette.**

4. **Type:**

   ```
   volcheck
   ```

5. **Copy the new boot environment's `boot/solaris/bootenv.rc` file to the diskette.**

   ```
   cp /a/boot/solaris/bootenv.rc /floppy/floppy0/solaris/bootenv.rc
   ```

6. **Check the input-device and output-device on the diskette to ensure that they are correct. If not, then update them.**

   You are ready to activate the new boot environment.

▼ To Activate a Boot Environment (Character Interface)

The first time you boot from a newly created boot environment, Solaris Live Upgrade software synchronizes the new boot environment with the boot environment that was last active. “Synchronize” means that certain critical system files and directories are copied from the last-active boot environment to the boot environment being booted. Solaris Live Upgrade does not perform this synchronization after this initial boot, unless you request to do so when prompted to force a synchronization.

For more information on synchronization, see “Synchronizing Files Between Boot Environments” on page 406.

1. **From the Solaris Live Upgrade main menu, select Activate.**
2. **Type the name of the boot environment to make active:**

   ```
   Name of Boot Environment: Solaris_9
   Do you want to force a Live Upgrade sync operations: no
   ```

3. **You can either continue or force a synchronization of files.**
   - Press Return to continue.

     The first time that the boot environment is booted, files are automatically synchronized.
   - You can force a synchronization of files, but use this feature with caution. Operating systems on each boot environment must be compatible with files that are being synchronized. To force a synchronization of files, type:

     ```
     Do you want to force a Live Upgrade sync operations: yes
     ```

   Caution – Use a forced synchronization with great care, because you might not be aware or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 9 software on your current boot environment and booted back to a Solaris 7 release with a forced synchronization, files could be changed on the Solaris 7 release. Because files are dependent on the release of the operating environment, the boot to the Solaris 7 release could fail because the Solaris 9 files might not be compatible with the Solaris 7 files.

4. **Press F3 to begin the activation process.**
5. **Press Return to continue.**

   The new boot environment is activated at the next reboot.
6. **To activate the inactive boot environment, reboot:**

```
# init 6
```

▼ To Activate a Boot Environment (Command-Line Interface)

1. **Log in as superuser.**
2. **To activate the boot environment, type:**

   ```
   # /usr/sbin/luactivate  BE_name
   ```

   BE_name Specifies the name of the boot environment that is to be activated
3. **Reboot.**

   ```
   # init 6
   ```

Caution – Use only the `init` or `shutdown` commands to reboot. If you use the `reboot`, `halt`, or `uadmin` commands, the system does not switch boot environments. The last active boot environment is booted again.

EXAMPLE 36–10 Activating a Boot Environment (Command-Line Interface)

In this example, the second_disk boot environment is activated at the next reboot.

```
# /usr/sbin/luactivate second_disk
# init 6
```

▼ To Activate a Boot Environment and Synchronize Files (Command-Line Interface)

The first time you boot from a newly created boot environment, Solaris Live Upgrade software synchronizes the new boot environment with the boot environment that was last active. “Synchronize” means that certain critical system files and directories are copied from the last-active boot environment to the boot environment being booted. Solaris Live Upgrade does not perform this synchronization after the initial boot, unless you force synchronization with the `luactivate` command and the `-s` option.

For more information on synchronization, see “Synchronizing Files Between Boot Environments” on page 406.

1. **Log in as superuser.**

2. **To activate the boot environment, type:**

```
# /usr/sbin/luactivate  -s BE_name
```

`-s` Forces a synchronization of files between the last-active boot environment and the new boot environment. The first time that a boot environment is activated, the files between the boot environment are synchronized With subsequent activations, the files are not synchronized unless you use the `-s` option.

> **Caution –** Use this option with great care, because you might not be aware or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 9 software on your current boot environment and booted back to a Solaris 7 release with a forced synchronization, files could be changed on the Solaris 7 release. Because files are dependent on the release of the operating environment, the boot to the Solaris 7 release could fail because the Solaris 9 files might not be compatible with the Solaris 7 files.

BE_name Specifies the name of the boot environment that is to be activated.

3. **Reboot.**

```
# init 6
```

EXAMPLE 36–11 Activating a Boot Environment (Command-Line Interface)

In this example, the second_disk boot environment is activated at the next reboot and the files are synchronized.

```
# /usr/sbin/luactivate -s second_disk
# init 6
```

Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)

If a failure is detected after upgrading or if the application is not compatible with an upgraded component, fall back to the original boot environment by using one of the following procedures, depending on your platform. For SPARC based systems, the first procedure runs `luactivate` to switch back to the original boot environment. If the first procedure fails, use the second procedure to boot from a DVD-ROM, CD-ROM, or net image. For x86 based systems, choose the procedure that is based on where the `root` (/) file systems are mounted, on the same or different physical disks.

▼ To Fall Back Despite Successful New Boot Environment Creation

Use this procedure when you have successfully activated your new boot environment, but are unhappy with the results.

1. **Log in as superuser.**

2. **Type:**

   ```
   # /usr/sbin/luactivate BE_name
   ```

   BE_name Specifies the name of the boot environment to be activated

3. **Reboot.**

   ```
   # init 6
   ```

   The previous working boot environment becomes the active boot environment.

▼ SPARC: To Fall Back From a Failed Boot Environment Activation

- If you experience a failure while booting the new boot environment and can boot the original boot environment in single-user mode, use this procedure to fall back to the original boot environment.
- If you need to boot from media or a net installation image, see "SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image" on page 467.

1. **At the OK prompt, boot the machine to single-user state from the Solaris DVD, Solaris Installation CD, the network, or a local disk.**

   ```
   OK boot cdrom -s
   ```

   or

   ```
   OK boot net -s
   ```

   or

   ```
   OK boot device_name -s
   ```

device_name Specifies the name of devices from where the system can boot, for example /dev/dsk/c0t0d0s0

2. **Type:**

```
# /sbin/luactivate BE_name
```

BE_name Specifies the name of the boot environment to be activated

- If this command fails to display a prompt, proceed to "SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image" on page 467.
- If the prompt is displayed, continue.

3. **At the prompt, type:**

```
Do you want to fallback to activate boot environment <disk name>
(yes or no)? yes
```

A message displays that the fallback activation is successful.

4. **Reboot.**

```
# init 6
```

The previous working boot environment becomes the active boot environment.

▼ SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image

Use this procedure to boot from a DVD, CD, or a net installation image. You need to mount the root (/) slice from the last active boot environment. Then run the luactivate command, which makes the switch. When you reboot, the last active boot environment is up and running again.

1. **At the OK prompt, boot the machine to single-user state from the Solaris DVD, Solaris Installation CD, the network, or a local disk:**

```
OK boot cdrom -s
```

or

```
OK boot net -s
```

or

```
OK boot device_name -s
```

device_name Specifies the name of the disk and the slice where a copy of the operating system resides, for example /dev/dsk/c0t0d0s0

2. **If necessary, check the integrity of the file system for the fallback boot environment root (/).**

```
# fsck device_name
```

device_name Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is entered in the form of `/dev/dsk/cwtxdysz`.

3. **Mount the active boot environment root (/) slice to some directory, such as `/mnt`:**

```
# mount device_name /mnt
```

device_name Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is entered in the form of `/dev/dsk/cwtxdysz`.

4. **From the active boot environment root (/) slice, type:**

```
# /mnt/sbin/luactivate
```

`luactivate` activates the previous working boot environment and indicates the result.

5. **Unmount `/mnt`**

```
# umount  /mnt
```

6. **Reboot.**

```
# init 6
```

The previous working boot environment becomes the active boot environment.

▼ x86: To Fall Back With Boot Environments on Different Disks

You might experience a failure while booting the new boot environment. If the `root` (/) file systems for the boot environments are on different physical disks, use the following procedure to fall back to the original boot environment.

1. **Reboot the machine and enter the appropriate BIOS menus.**
   - If your boot devices are SCSI, refer to documentation on your SCSI controller on how to enter the SCSI BIOS.
   - If the boot devices are maintained by the system BIOS, refer to system BIOS documentation on how to enter the system BIOS.
2. **Follow the appropriate BIOS documentation to change the boot device back to the original boot environment's boot device if different.**

3. **Save the BIOS changes.**

4. **Exit BIOS to begin the boot process.**

5. **Type `b -s` to boot the machine to single-user state.**

6. **Type:**

   ```
   # /sbin/luactivate
   ```

7. **Reboot.**

   ```
   # init 6
   ```

▼ x86: To Fall Back With Boot Environments on the Same Disk

You might experience a failure while booting. If the `root` (/) file systems are on the same physical disk, use the following procedure to fall back to the original boot environment. You need to mount the `root` (/) slice from the last active boot environment. Then run the `luactivate` command, which makes the switch. When you reboot, the last active boot environment is up and running again.

1. **Decide how to boot the system.**

   - If you boot from the Solaris DVD or the Solaris Installation CD, insert the disc. Your system's BIOS must support booting from a DVD or CD.
   - If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the systems's BIOS setup tool or the network adapter's configuration setup Tool.
   - If you boot from a diskette, insert Solaris Device Configuration Assistant *x86 Platform Edition* diskette into the system's diskette drive.

     x86 only – You can copy the Device Configuration Assistant software to a diskette from the Solaris *x86 Platform Edition* DVD or Solaris Software 2 of 2 *x86 Platform Edition* CD by using the procedure described in Appendix I.

   Follow the directions on screen until the Current Boot Parameters menu is displayed.

2. **Type `b -s` to boot the machine to single-user state.**

3. **If necessary, check the integrity of the file system for the fallback boot environment `root` (/).**

```
# fsck mount_ point
```

mount_point A root file system that is known and reliable

4. **Mount the active boot environment root slice to some directory (such as `/mnt/sbin`):**

```
# mount device_name /mnt/sbin
```

device_name Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is entered in the form of /dev/dsk/c*w*t*x*d*y*s*z*.

5. **From the active boot environment root slice, type:**

```
# /mnt/sbin/luactivate
```

`luactivate` activates the previous working boot environment and indicates the result.

6. **Unmount `/mnt/sbin`.**

```
# umount device_name /mnt/sbin
```

device_name Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is entered in the form of /dev/dsk/c*w*t*x*d*y*s*z*.

7. **Reboot.**

```
# init 6
```

The previous working boot environment becomes the active boot environment.

CHAPTER 37

Maintaining Solaris Live Upgrade Boot Environments (Tasks)

This chapter explains various maintenance tasks such as keeping a boot environment file system up to date or deleting a boot environment. This chapter contains the following sections:

- "Overview of Solaris Live Upgrade Maintenance" on page 471
- "Displaying the Status of All Boot Environments" on page 472
- "Updating a Previously Configured Boot Environment" on page 474
- "Canceling a Scheduled Create, Upgrade, or Copy Job" on page 476
- "Comparing Boot Environments" on page 477
- "Deleting an Inactive Boot Environment" on page 478
- "Displaying the Name of the Active Boot Environment" on page 479
- "Changing the Name of a Boot Environment" on page 480
- "Adding or Changing a Description Associated With a Boot Environment Name" on page 482
- "Viewing the Configuration of a Boot Environment" on page 484

Overview of Solaris Live Upgrade Maintenance

TABLE 37–1 Overview of Solaris Live Upgrade Maintenance

Task	Description	For Instructions
(Optional) View Status	■ View whether a boot environment is active, being activated, scheduled to be activated, or in the midst of a comparison	■ "Displaying the Status of All Boot Environments" on page 472

TABLE 37–1 Overview of Solaris Live Upgrade Maintenance *(Continued)*

Task	Description	For Instructions
	■ Compare the active and inactive boot environments ■ Display the name of the active boot environment ■ View the configurations of a boot environment	■ "Comparing Boot Environments" on page 477 ■ "Displaying the Name of the Active Boot Environment" on page 479 ■ "Viewing the Configuration of a Boot Environment" on page 484
(Optional) Update an inactive boot environment	Copy file systems from the active boot environment again without changing the configuration of file systems	"Updating a Previously Configured Boot Environment" on page 474
(Optional) Other tasks	■ Delete a boot environment ■ Change the name of a boot environment ■ Add or change a description that is associated with a boot environment name ■ Cancel scheduled jobs	■ "Deleting an Inactive Boot Environment" on page 478 ■ "Changing the Name of a Boot Environment" on page 480 ■ "Adding or Changing a Description Associated With a Boot Environment Name" on page 482 ■ "Canceling a Scheduled Create, Upgrade, or Copy Job" on page 476

Displaying the Status of All Boot Environments

Use the Status menu or the `lustatus` command to display the information about the boot environment. If no boot environment is specified, the status information for all boot environments on the system is displayed.

The following details for each boot environment are displayed:

- Name – Name of each boot environment.
- Complete – Indicates that no copy or create operations are in progress. Also, the boot environment can be booted. Any current activity or failure in a create or upgrade operation causes a boot environment to be incomplete. For example, if a copy operation is in process or scheduled for a boot environment, that boot

environment is considered incomplete.

- Active – Indicates if this is the active boot environment.
- ActiveOnReboot – Indicates if the boot environment becomes active on next reboot of the system.
- CopyStatus – Indicates if the creation or copy of the boot environment is scheduled, active, or in the process of being upgraded. A status of `SCHEDULED` prevents you performing live upgrade copy, rename, or upgrade operations.

To Display the Status of All Boot Environments (Character Interface)

- **From the main menu, select Status.**

  A table that is similar to the following is displayed:

```
boot environment   Is        Active  Active     Can        Copy
Name               Complete  Now     OnReboot   Delete     Status
------------------------------------------------------------------------
disk_a_S7          yes       yes     yes        no       -
disk_b_S7database  yes       no      no         yes      COPYING
disk_b_S8          no        no      no         yes      -
```

Note – In this example, you could not perform copy, rename, or upgrade operations on `disk_b_S8` because it is not complete, nor on `disk_b_S7database`, because a live upgrade operation is in progress.

To Display the Status of All Boot Environments (Command-Line Interface)

1. **Log in as superuser.**
2. **Type:**

```
# lustatus BE_name
```

BE_name Specifies the name of the inactive boot environment to view status. If *BE_name* is omitted, `lustatus` displays status for all boot environments in the system.

In this example, the status for all boot environments is displayed.

```
# lustatus second_disk
boot environment   Is        Active  Active     Can        Copy
Name               Complete  Now     OnReboot   Delete     Status
------------------------------------------------------------------------
```

```
disk_a_S7          yes       yes     yes       no       -
disk_b_S7database  yes       no      no        yes      COPYING
disk_b_S8          no        no      no        yes      -
```

Note – You could not perform copy, rename, or upgrade operations on `disk_b_S8` because it is not complete, nor on `disk_b_S7database` because a live upgrade operation is in progress.

Updating a Previously Configured Boot Environment

You can update the contents of a previously configured boot environment with the Copy menu or the `lumake` command. File Systems from the active (source) boot environment are copied to the target boot environment. The data on the target is also destroyed. A boot environment must have the status "complete" before you can copy from it. See "Displaying the Status of All Boot Environments" on page 472 to determine a boot environment's status.

The copy job can be scheduled for a later time, and only one job can be scheduled at a time. To cancel a scheduled copy, see "Canceling a Scheduled Create, Upgrade, or Copy Job" on page 476.

▼ To Update a Previously Configured Boot Environment (Character Interface)

1. **From the main menu, select Copy.**
2. **Type the name of the inactive boot environment to update:**

   ```
   Name of Target Boot Environment: solaris8
   ```

3. **Continue or schedule the copy to occur later:**
   - To continue with the copy, press Return.

     The inactive boot environment is updated.
   - To schedule the copy for later, type **`y`**, a time (by using the `at` command format), and the email address to which to send the results:

```
Do you want to schedule the copy? y
Enter the time in 'at' format to schedule copy: 8:15 PM
Enter the address to which the copy log should be mailed:
someone@anywhere.com
```

For information on time formats, see the `at`(1) man page.

The inactive boot environment is updated.

To cancel a scheduled copy, see "Canceling a Scheduled Create, Upgrade, or Copy Job" on page 476.

▼ To Update a Previously Configured Boot Environment (Command-Line Interface)

This procedure copies source files over outdated files on a boot environment that was previously created.

1. **Log in as superuser.**
2. **Type:**

```
# lumake -n  BE_name [-s source_BE] [-t  time] [-m email_address]
```

`-n` *BE_name* Specifies the name of the boot environment that has file systems that are to be replaced.

`-s` *source_BE* (Optional) Specifies the name of the source boot environment that contains the file systems to be copied to the target boot environment. If you omit this option, `lumake` uses the current boot environment as the source.

`-t` *time* (Optional) Set up a batch job to copy over file systems on a specified boot environment at a specified time. The time is given in the format that is specified by the man page, `at`(1).

`-m` *email_address* (Optional) Enables you to email `lumake` output to a specified address on command completion. *email_address* is not checked. You can use this option only in conjunction with `-t`.

EXAMPLE 37–1 Updating a Previously Configured Boot Environment (Command-Line Interface)

In this example, file systems from `first_disk` are copied to `second_disk`. When the job is completed, an email is sent to Joe at `anywhere.com`.

```
# lumake -n  second_disk -s first_disk -m joe@anywhere.com
```

EXAMPLE 37–1 Updating a Previously Configured Boot Environment (Command-Line Interface) *(Continued)*

The files on `first_disk` are copied to `second_disk` and email is sent for notification. To cancel a scheduled copy, see "Canceling a Scheduled Create, Upgrade, or Copy Job" on page 476.

Canceling a Scheduled Create, Upgrade, or Copy Job

A boot environment's scheduled creation, upgrade, or copy job can be canceled just prior to the time the job starts. A job can be scheduled for a specific time either in the GUI with the Create a Boot Environment, Upgrade a Boot Environment, or Copy a Boot Environment menus. In the CLI, the job can be scheduled by the `lumake` command. At any time, only one job can be scheduled on a system.

▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Character Interface)

1. **From the main menu, select Cancel.**
2. **To view a list of boot environments that is available for canceling, press F2.**
3. **Select the boot environment to cancel.**

   The job no longer executes at the time specified.

▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Command-Line Interface)

1. **Log in as superuser.**
2. **Type:**

   ```
   # lucancel
   ```

   The job no longer executes at the time that is specified.

Comparing Boot Environments

Use the Compare menu or `lucompare` to check for differences between the active boot environment and other boot environments. To make a comparison, the inactive boot environment must be in a complete state and cannot have a copy job that is pending. See “Displaying the Status of All Boot Environments” on page 472.

The specified boot environment cannot have any partitions that are mounted with `lumount` or `mount`.

▼ To Compare Boot Environments (Character Interface)

1. **From the main menu, select Compare.**

2. **Select either Compare to Original or Compare to an Active Boot Environment.**

3. **Press F3.**

4. **Type the names of the original (active) boot environment, the inactive boot environment, and the path to a file:**

   ```
   Name of Parent: solaris8
   Name of Child: solaris8-1
   Full Pathname of the file to Store Output: /tmp/compare
   ```

5. **To save to the file, press F3.**

   The Compare menu displays the following file attributes:

   - Mode.
   - Number of links.
   - Owner.
   - Group.
   - Checksum – Computes checksums only if the file in the specified boot environment matches its counterpart on the active boot environment in all of the fields that are described previously. If everything matches but the checksums differ, the differing checksums are appended to the entries for the compared files.
   - Size.
   - Existence of files in only one boot environment.

6. **To return to the Compare menu, press F3.**

▼ To Compare Boot Environments (Command-Line Interface)

1. **Log in as superuser.**
2. **Type:**

```
# /usr/sbin/lucompare -i  infile (or) -t -o  outfile BE_name
```

`-i` *infile* Compare files that are listed in *infile*. The files to be compared should be an absolute file name. If the entry in the file is a directory, then comparison is recursive to the directory. Use either this option or `-t`, not both.

`-t` Compare only nonbinary files. This comparison uses the `file`(1) command on each file to determine if the file is a text file. Use either this option or `-i`, not both.

`-o` *outfile* Redirect the output of differences to *outfile*.

BE_name Specifies the name of the boot environment that is compared to the active boot environment.

EXAMPLE 37–2 Comparing Boot Environments (Command-Line Interface)

In this example, first_disk boot environment (source) is compared to second_disk boot environment and the results are sent to a file.

```
# /usr/sbin/lucompare -i  /etc/lu/compare/ \
-o /var/tmp/compare.out second_disk
```

Deleting an Inactive Boot Environment

Use either the Delete menu or `ludelete`. You cannot delete the active boot environment or the boot environment that is activated on the next reboot. The boot environment to be deleted must be complete. A complete boot environment is not participating in an operation that will change its status. Use “Displaying the Status of All Boot Environments” on page 472 to determine a boot environment’s status. Also, you cannot delete a boot environment that has file systems mounted with `lumount`.

▼ To Delete an Inactive Boot Environment (Character Interface)

1. **From the main menu, select Delete.**
2. **Type the name of the inactive boot environment you want to delete:**

   ```
   Name of boot environment: solaris8
   ```

   The inactive boot environment is deleted.

▼ To Delete an Inactive Boot Environment (Command-Line Interface)

1. **Log in as superuser.**
2. **Type:**

   ```
   # ludelete BE_name
   ```

   BE_name Specifies the name of the inactive boot environment that is to be deleted

EXAMPLE 37–3 Deleting an Inactive Boot Environment (Command-Line Interface)

In this example, the boot environment, second_disk, is deleted.

```
# ludelete second_disk
```

Displaying the Name of the Active Boot Environment

Use the Current menu or the `lucurr` command to display the name of the currently running boot environment. If no boot environments are configured on the system, the message "No Boot Environments are defined" is displayed. Note that `lucurr` reports only the name of the current boot environment, not the boot environment that is active on the next reboot. See "Displaying the Status of All Boot Environments" on page 472 to determine a boot environment's status.

▼ To Display the Name of the Active Boot Environment (Character Interface)

- **From the main menu, select Current.**

  The active boot environment's name or the message "No Boot Environments are defined" is displayed.

▼ To Display the Name of the Active Boot Environment (Command-Line Interface)

- **Type:**

  ```
  # /usr/sbin/lucurr
  ```

EXAMPLE 37–4 Displaying the Name of the Active Boot Environment (Command-Line Interface)

In this example, the name of the current boot environment is displayed.

```
# /usr/sbin/lucurr
solaris8
```

Changing the Name of a Boot Environment

Renaming a boot environment is often useful when you upgrade the boot environment from one Solaris release to another release. For example, following an operating system upgrade, you might rename the boot environment `solaris7` to `solaris8`. Use the Rename menu or `lurename` command to change the inactive boot environment's name.

The new name must adhere to the following limitations.

- Not exceed 30 characters in length.
- Consist only of alphanumeric characters and other ASCII characters that are not special to the UNIX shell. See the "Quoting" section of `sh`(1).
- Contain only single-byte, 8–bit characters.
- Be unique on the system.

A boot environment must have the status "complete" before you rename it. See "Displaying the Status of All Boot Environments" on page 472 to determine a boot environment's status. You cannot rename a boot environment that has file systems mounted with `lumount` or `mount`.

▼ To Change the Name of an Inactive Boot Environment (Character Interface)

1. **From the main menu, select Rename.**
2. **Type the boot environment to rename and then the new name.**
3. **To save your changes, press F3.**

▼ To Change the Name of an Inactive Boot Environment (Command-Line Interface)

1. **Log in as superuser.**
2. **Type:**

   ```
   # lurename -e  BE_name -n  new_name
   ```

`-e` *BE_name*	Specifies the inactive boot environment name to be changed
`-n` *new_name*	Specifies the new name of the inactive boot environment

   In this example, second_disk is renamed to third_disk.

   ```
   # lurename -e  second_disk  -n  third_disk
   ```

Adding or Changing a Description Associated With a Boot Environment Name

You can associate a description with a boot environment name. The description never replaces the name. Although a boot environment name is restricted in length and characters, the description can be of any length and of any content. The description can be simple text or as complex as a `gif` file. You can create this description at these times:

- When you create a boot environment with the `lucreate` command and use the `-A` option
- After the boot environment has been created by using the `ludesc` command

For more information on using the `-A` option with `lucreate`, see "To Create a Boot Environment for the First Time (Command-Line Interface)" on page 419. For more information on creating the description after the boot environment has been created, see the following procedures or the man page, `ludesc`(1M).

▼ To Add or Change a Description for a Boot Environment Name With Text

1. **Log in as superuser.**

2. **Type:**

```
# /usr/sbin/ludesc -n  BE_name  'BE_description'
```

`-n` *BE_name 'BE_description'* — Specifies the boot environment name and the new description to be associated with the name

EXAMPLE 37–5 Adding a Description to a Boot Environment Name With Text

In this example, a boot environment description is added to a boot environment that is named second_disk. The description is text that is enclosed in single quotes.

```
# /usr/sbin/ludesc -n second_disk 'Solaris 9 test build'
```

▼ To Add or Change a Description for a Boot Environment Name With a File

1. **Log in as superuser.**
2. **Type:**

```
# /usr/sbin/ludesc -n BE_name  -f file_name
```

-n *BE_name* Specifies the boot environment name

file_name Specifies the file to be associated with a boot environment name

EXAMPLE 37–6 Adding a Description to a Boot Environment Name With a File

In this example, a boot environment description is added to a boot environment that is named second_disk. The description is contained in a `gif` file.

```
# /usr/sbin/ludesc -n second_disk -f rose.gif
```

▼ To Determine a Boot Environment Name From a Text Description

1. **Log in as superuser.**
2. **Type:**

```
# /usr/sbin/ludesc -A 'BE_description'
```

-A *'BE_description'* Displays the boot environment name that is associated with the description

EXAMPLE 37–7 Determining a Boot Environment Name From a Description

In this example, the name of the boot environment, second_disk, is determined by using the `-A` option with the description.

```
# /usr/sbin/ludesc -A  'Solaris 9 test build'
 second_disk
```

▼ To Determine a Boot Environment Name From a Description in a File

1. **Log in as superuser.**
2. **Type:**

```
# /usr/sbin/ludesc -f  file_name
```

-f *file_name* Displays the boot environment name that is associated with the description that is located in a file

EXAMPLE 37–8 Determining a Boot Environment Name From a Description in a File

In this example, the name of the boot environment, second_disk, is determined by using the -f option and the name of the file that contains the description.

```
# /usr/sbin/ludesc -f rose.gif
second_disk
```

▼ To Determine a Boot Environment Description From a Name

1. **Log in as superuser.**

2. **Type:**

```
# /usr/sbin/ludesc -n BE_name
```

-n *BE_name* Displays the description that is associated with the boot environment name

EXAMPLE 37–9 Determining a Boot Environment Description From a Name

In this example, the description is determined by using the -n option with the boot environment name.

```
# /usr/sbin/ludesc -n  second_disk
Solaris 9 test build
```

Viewing the Configuration of a Boot Environment

Use the List menu or the lufslist command to list the configuration of a boot environment. The output contains the disk slice (file system), file system type, and file system size for each boot environment mount point.

▼ To View the Configuration of Each Inactive Boot Environment (Character Interface)

1. **From the main menu, select List.**

2. **To view the status of a boot environment, type the name.**

   ```
   Name of Boot Environment: solaris8
   ```

3. **Press F3.**

   The following example displays a list.

   ```
   Filesystem                fstype       size(Mb) Mounted on
   ------------------------------------------------------------------
   /dev/dsk/c0t0d0s1         swap           512.11 -
   /dev/dsk/c0t4d0s3         ufs           3738.29 /
   /dev/dsk/c0t4d0s4         ufs            510.24 /opt
   ```

4. **To return to the List menu, press F6.**

▼ To View the Configuration of a Boot Environment (Command-Line Interface)

1. **Log in as superuser.**

2. **Type:**

   ```
   # lufslist
   ```

   BE_name Specifies the name of the boot environment to view file system specifics

   The following example displays a list.

   ```
   Filesystem                fstype       size(Mb) Mounted on
   ------------------------------------------------------------------
   /dev/dsk/c0t0d0s1         swap           512.11 -
   /dev/dsk/c0t4d0s3         ufs           3738.29 /
   /dev/dsk/c0t4d0s4         ufs            510.24 /opt
   ```

CHAPTER **38**

Solaris Live Upgrade (Examples)

This chapter provides examples of creating a boot environment, then upgrading and activating the new boot environment which then becomes the currently running system. This chapter contains the following sections:

Example of Upgrading With Solaris Live Upgrade (Command-Line Interface)

In this example, a new boot environment is created by using the `lucreate` command on a system that is running the Solaris 7 release. The new boot environment is upgraded to the Solaris 9 release by using the `luupgrade` command. The upgraded boot environment is activated by using the `luactivate` command. An example of falling back to the original boot environment is also given.

Install Live Upgrade on the Active Boot Environment

1. Insert the Solaris DVD or Solaris Software 2 of 2 CD.
2. Follow the step for the media you are using.
   - If you are using the Solaris DVD, change directories to the installer and run the installer.

     ```
     # cd /cdrom/cdrom0/Solaris_9/Tools/Installers
     # ./liveupgrade20
     ```

     The Solaris Web Start installer is displayed.
   - If you are using the Solaris Software 2 of 2 CD, run the installer.

     ```
     % ./installer
     ```

     The Solaris Web Start installer is displayed.
3. From the Select Type of Install panel, click Custom.
4. On the Locale Selection panel, click the language to be installed.
5. Choose the software to install.
   - For DVD, on the Component Selection panel, click Next to install the packages.
   - For CD, On the Product Selection panel, click Default Install for Solaris Live Upgrade and click on the other product choices to deselect this software.
6. Follow the directions on the Solaris Web Start installer panels to install the software.

Create a Boot Environment

The source boot environment is named c0t4d0s0 by using the `-c` option. Naming the source boot environment is only required when the first boot environment is created. For more information on naming using the `-c` option, see the description in Step 2.

The new boot environment is named c0t15d0s0. The `-A` option creates a description that is associated with the boot environment name.

The root (`/`) file system is copied to the new boot environment. Also, a new swap slice is created rather than sharing the source boot environment's swap slice.

```
# lucreate -A 'BE_description' -c c0t4d0s0 -m /:/dev/dsk/c0t15d0s0:ufs \
-m -:/dev/dsk/c0t15d0s1:swap -n c0t15d0s0
```

Upgrade the Inactive Boot Environment

The inactive boot environment is named c0t15d0s0. The operating system image to be used for the upgrade is taken from the network.

```
# luupgrade -n c0t15d0s0 -u -s /net/ins-svr/export/Solaris_9 \
combined.solaris_wos
```

Check If Boot Environment Is Bootable

The `lustatus` command reports if the boot environment creation is complete. `lustatus` also shows if the boot environment is bootable.

```
# lustatus
boot environment   Is        Active  Active     Can        Copy
Name               Complete  Now     OnReboot   Delete     Status
------------------------------------------------------------------------
c0t4d0s0           yes       yes      yes      no      -
c0t15d0s0          yes       no       no       yes     -
```

Activate the Inactive Boot Environment

The c0t15d0s0 boot environment is made bootable with the `luactivate` command. The system is then rebooted and c0t15d0s0 becomes the active boot environment. The c0t4d0s0 boot environment is now inactive.

```
# luactivate c0t15d0s0
# init 6
```

Fall Back to the Source Boot Environment

Three procedures for falling back depend on your new boot environment activation situation:

- The activation is successful, but you want to return to the original boot environment. See Example 38–1.
- The activation fails and you can boot back to the original boot environment. See Example 38–2.
- The activation fails and you must boot back to the original boot environment by using media or a net installation image. See Example 38–3.

EXAMPLE 38–1 To Fall Back Despite Successful Boot Environment Creation

In this example, the original c0t4d0s0 boot environment is reinstated as the active boot environment although it was activated successfully. The device name is first_disk.

EXAMPLE 38–1 To Fall Back Despite Successful Boot Environment Creation *(Continued)*

```
# /usr/sbin/luactivate first_disk
# init 6
```

EXAMPLE 38–2 SPARC: To Fall Back From a Failed Boot Environment Activation

In this example, the new boot environment was not bootable. You must return to the OK prompt before booting from the original boot environment, c0t4d0s0, in single-user mode.

```
OK boot net -s
# /sbin/luactivate first_disk
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# init 6
```

The original boot environment, c0t4d0s0, becomes the active boot environment.

EXAMPLE 38–3 SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image

In this example, the new boot environment was not bootable. You cannot boot from the original boot environment and must use media or a net installation image. The device is `/dev/dsk/c0t4d0s0`. The original boot environment, c0t4d0s0, becomes the active boot environment.

```
OK boot net -s
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

Example of Detaching and Upgrading One Side of a RAID 1 Volume (Mirror) (Command-Line Interface)

This example shows you how to do the following tasks:

- Create a RAID-1 volume (mirror) on a new boot environment
- Break the mirror and upgrade one half of the mirror
- Attach the other half of the mirror, the concatenation, to the new mirror

Figure 38–1 shows the current boot environment that contains three physical disks.

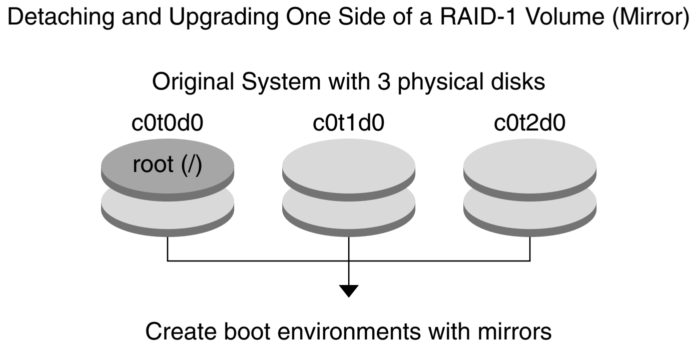

Detaching and Upgrading One Side of a RAID-1 Volume (Mirror)
Original System with 3 physical disks
c0t0d0
c0t1d0
c0t2d0
root (/)
Create boot environments with mirrors

FIGURE 38–1 Detaching and Upgrading One Side of a RAID-1 Volume (Mirror)

1. Create a new boot environment, `second_disk`, that contains a mirror.

   The following command performs these tasks.

   - `lucreate` configures a UFS file system for the mount point root (`/`). A mirror, `d10`, is created. This mirror is the receptacle for the current boot environment's root (`/`) file system that is copied to the mirror `d10`. All data on the mirror `d10` is overwritten.
   - Two slices, `c0t1d0s0` and `c0t2d0s0`, are specified to be used as submirrors. These two submirrors are attached to mirror `d10`.

   ```
   # lucreate -c first_disk -n second_disk \
   -m /:/dev/md/dsk/d10:ufs,mirror \
   -m /:/dev/dsk/c0t1d0s0:attach \
   -m /:/dev/dsk/c0t2d0s0:attach
   ```

2. Activate the `second_disk` boot environment.

   ```
   # /usr/sbin/luactivate second_disk
   # init 6
   ```

3. Create another boot environment, `third_disk`.

   The following command performs these tasks.

   - `lucreate` configures a UFS file system for the mount point root (`/`). A mirror, `d20`, is created.
   - Slice `c0t1d0s0` is removed from its current mirror and is added to mirror `d20`. The contents of the submirror, the root (`/`) file system, are preserved and no copy occurs.

   ```
   # lucreate -n third_disk \
   -m /:/dev/md/dsk/d20:ufs,mirror \
   -m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
   ```

4. Upgrade the new boot environment, `third_disk`, by installing a Solaris Flash archive. The archive is located on the local system. The operating systems versions for the `-s` and `-a` options are both Solaris 9 releases. All files are overwritten on `third_disk` except shareable files.

   ```
   # luupgrade -f -n third_disk \
   -s /net/installmachine/export/Solaris_9/OS_image \
   -a /net/server/archive/Solaris_9
   ```

5. Activate the `third_disk` boot environment to make this boot environment the currently running system.

   ```
   # /usr/sbin/luactivate third_disk
   # init 6
   ```

6. Delete the boot environment `second_disk`.

   ```
   # ludelete second_disk
   ```

7. The following commands perform these tasks.

   - Clear mirror `d10`.
   - Check for the number for the concatenation of `c0t2d0s0`.
   - Attach the concatenation that is found by the `metastat` command to the mirror `d20`. The `metattach` command synchronizes the newly attached concatenation with the concatenation in mirror `d20`. All data on the concatenation is overwritten.

   ```
   # metaclear d10
   metastat -p | grep c0t2d0s0
   dnum 1 1 c0t2d0s0
   metattach d20 dnum
   ```

   num Is the number found in the `metastat` command for the concatenation

The new boot environment, `third_disk`, has been upgraded and is the currently running system. `third_disk` contains the root (/) file system that is mirrored.

Figure 38–2 shows the entire process of detaching a mirror and upgrading the mirror by using the commands in the preceding example.

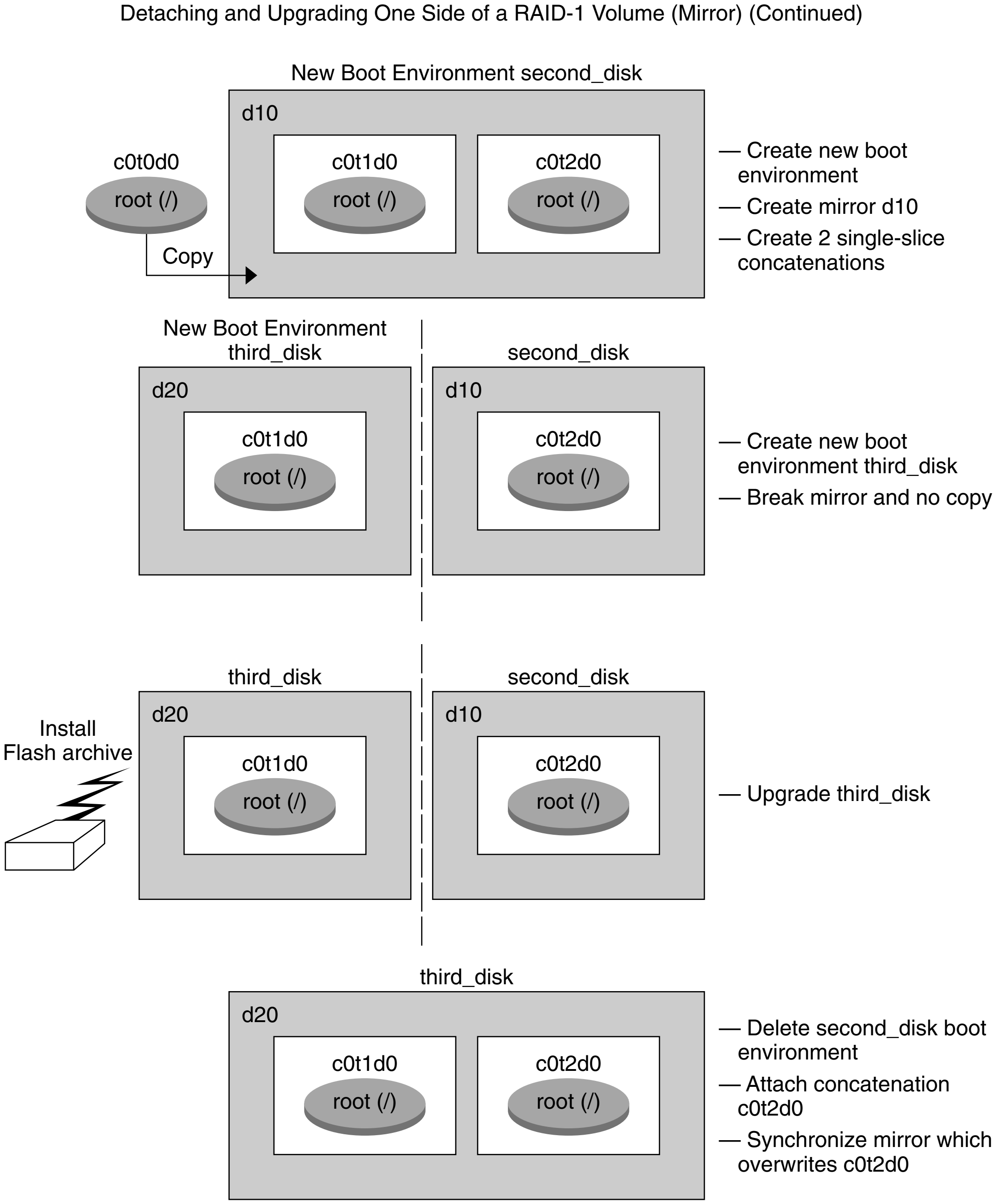

Detaching and Upgrading One Side of a RAID-1 Volume (Mirror) (Continued)
New Boot Environment second_disk
d10
c0t0d0
root (/)
c0t1d0
root (/)
c0t2d0
root (/)
Copy
— Create new boot environment
— Create mirror d10
— Create 2 single-slice concatenations
New Boot Environment third_disk
second_disk
d20
c0t1d0
root (/)
d10
c0t2d0
root (/)
— Create new boot environment third_disk
— Break mirror and no copy
third_disk
second_disk
Install Flash archive
d20
c0t1d0
root (/)
d10
c0t2d0
root (/)
— Upgrade third_disk
third_disk
d20
c0t1d0
root (/)
c0t2d0
root (/)
— Delete second_disk boot environment
— Attach concatenation c0t2d0
— Synchronize mirror which overwrites c0t2d0

FIGURE 38–2 Detaching and Upgrading One Side of a RAID-1 Volume (Mirror) (continued)

Example of Migrating From an Existing Volume to a Solaris Volume Manager RAID-1 Volume (Command-Line Interface)

Solaris Live Upgrade enables the creation of a new boot environment on RAID–1 volumes (mirrors). The current boot environment's file systems can be on any of the following:

- A physical storage device
- A Solaris Volume Manager controlled RAID–1 volume
- A Veritas VXFS controlled volume

However, the new boot environment's target must be a Solaris Volume Manager RAID-1 volume. For example, the slice that is designated for a copy of the root (/) file system must be `/dev/md/dsk/rootvol`. `rootvol` is the volume that contains the root (/) file system.

In this example, the current boot environment contains the root (/) file system on a volume that is not a Solaris Volume Manager volume. The new boot environment is created with root (/) on the Solaris Volume Manager RAID-1 volume c0t2d0s0. The `lucreate` command migrates the current volume to the Solaris Volume Manager volume. The name of the new boot environment is `svm_be`. The `lustatus` command reports if the new boot environment is ready to be activated and be rebooted. The new boot environment is activated to become the current boot environment.

```
# lucreate -n svm_be -m /:/dev/md/dsk/d1:mirror,ufs \
 -m /:/dev/dsk/c0t2d0s0:attach
# lustatus
# luactivate svm_be
# lustatus
# init 6
```

Example of Creating an Empty Boot Environment and Installing a Solaris Flash Archive (Command-Line Interface)

The following procedures take you through the three step process:

- Creating the empty boot environment
- Installing the archive
- Activating the boot environment which then becomes the currently running boot environment.

The `lucreate` command creates a boot environment that is based on the file systems in the active boot environment. When using the `lucreate` command with the `-s -` option, `lucreate` quickly creates an empty boot environment. The slices are reserved for the file systems specified, but no file systems are copied. The boot environment is named, but not actually created until installed with a Solaris Flash archive. When the empty boot environment is installed with an archive, file systems are installed on the reserved slices. The boot environment is then activated.

Creating an Empty Boot Environment

In this first step, an empty boot environment is created. Slices are reserved for the file systems specified, but no copy of file systems from the current boot environment occurs. The new boot environment is named `second_disk`.

```
# lucreate  -s - -m /:/dev/dsk/c0t1d0s0:ufs \
 -n second_disk
```

The boot environment is ready to be populated with a Solaris Flash archive.

Figure 38–3 shows the creation of an empty boot environment.

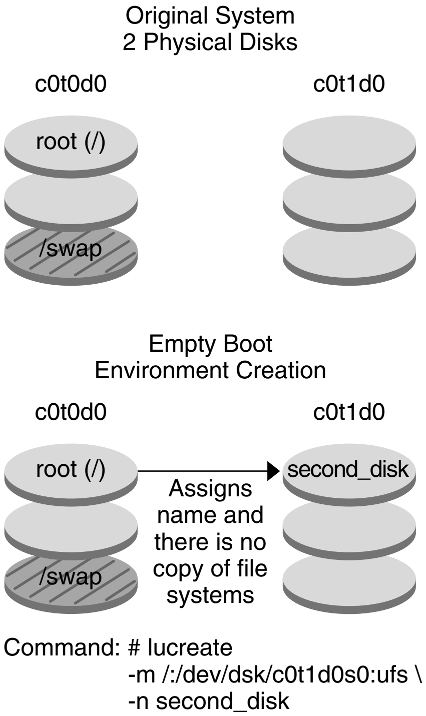


FIGURE 38–3 Creating an Empty Boot Environment

Installing the Solaris Flash Archive on the New Boot Environment

In this second step, an archive is installed on the `second_disk` boot environment that was created in the above example. The archive is located on the local system. The operating system versions for the `-s` and `-a` options are both Solaris 9 releases. The archive is named `Solaris_9.flar`.

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_9/OS_image \
-a /net/server/archive/Solaris_9.flar
```

The boot environment is ready to be activated.

Activating the New Boot Environment

In this last step, the `second_disk` boot environment is made bootable with the `luactivate` command. The system is then rebooted and `second_disk` becomes the active boot environment.

```
# luactivate second_disk
# init 6
```

- For step-by-step information about creating an empty boot environment, see "To Create an Empty Boot Environment for a Solaris Flash Archive (Command-Line Interface)" on page 432.
- For step-by-step information about creating a Solaris Flash archive, see Chapter 21.
- For step-by-step information about activating a boot environment or falling back to the original boot environment, see "Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)" on page 465.

Example of Upgrading Using Solaris Live Upgrade (Character Interface)

In this example, a new boot environment is created on a system that is running the Solaris 7 release. The new boot environment is upgraded to the Solaris 9 release. The upgraded boot environment is then activated.

Install Live Upgrade on the Active Boot Environment

1. Insert the Solaris DVD or Solaris Software 2 of 2 CD.
2. Run the installer for the media you are using.
   - If you are using the Solaris DVD, change directories to the installer and run the installer.

     ```
     # cd /cdrom/cdrom0/Solaris_9/Tools/Installers
     # ./liveupgrade20
     ```

     The Solaris Web Start installer is displayed.
   - If you are using the Solaris Software 2 of 2 CD, run the installer.

     ```
     % ./installer
     ```

The Solaris Web Start installer displays.

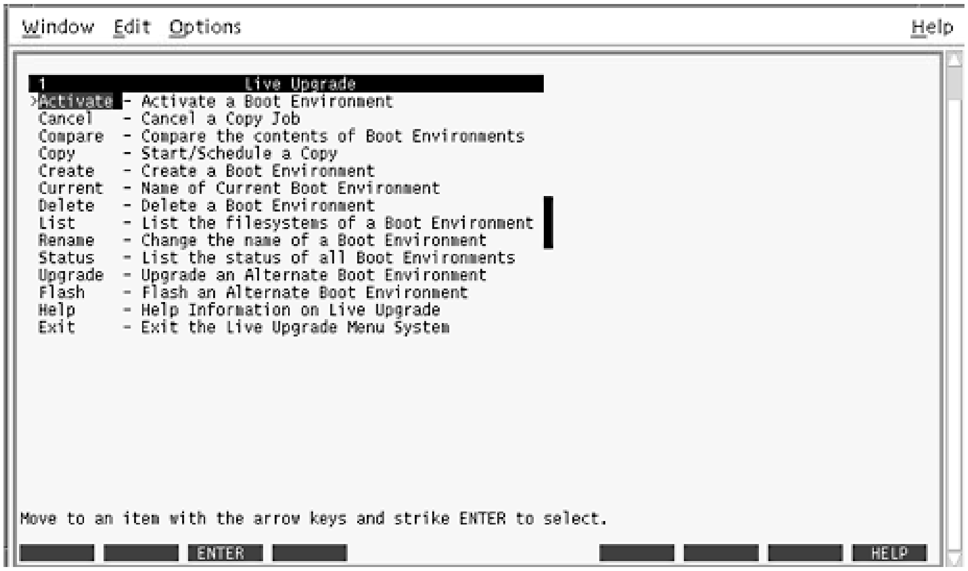
3. From the Select Type of Install panel, click Custom.
4. On the Locale Selection panel, click the language to be installed.
5. Choose the software to install.
   - For DVD, on the Component Selection panel, click Next to install the packages.
   - For CD, On the Product Selection panel, click Default Install for Solaris Live Upgrade and click on the other product choices to deselect the software.
6. Follow the directions on the Solaris Web Start installer panels to install the software.

Create a Boot Environment

In this example, the source boot environment is named c0t4d0s0. The root (/) file system is copied to the new boot environment. Also, a new swap slice is created rather than sharing the source boot environment's swap slice.

1. Display the character interface:

```
# /usr/sbin/lu
```



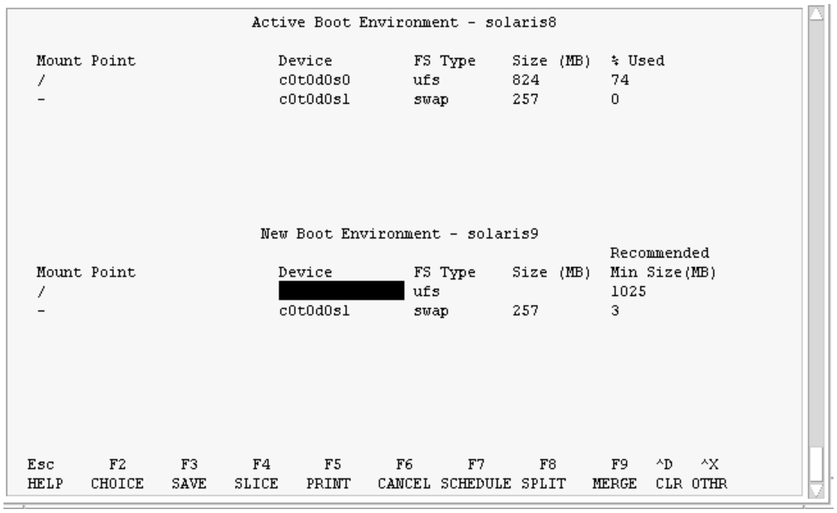
```
1                     Live Upgrade
>Activate  - Activate a Boot Environment
 Cancel    - Cancel a Copy Job
 Compare   - Compare the contents of Boot Environments
 Copy      - Start/Schedule a Copy
 Create    - Create a Boot Environment
 Current   - Name of Current Boot Environment
 Delete    - Delete a Boot Environment
 List      - List the filesystems of a Boot Environment
 Rename    - Change the name of a Boot Environment
 Status    - List the status of all Boot Environments
 Upgrade   - Upgrade an Alternate Boot Environment
 Flash     - Flash an Alternate Boot Environment
 Help      - Help Information on Live Upgrade
 Exit      - Exit the Live Upgrade Menu System

Move to an item with the arrow keys and strike ENTER to select.
```

FIGURE 38–4 Solaris Live Upgrade Main Menu

2. From the main menu, select Create.

```
Name of Current Boot Environment:    c0t4d0s0
Name of New Boot Environment:   c0t15d0s0
```

3. Press F3.



```
                         Active Boot Environment - solaris8

  Mount Point                Device        FS Type    Size (MB)  % Used
  /                          c0t0d0s0      ufs        824        74
  -                          c0t0d0s1      swap       257        0





                          New Boot Environment - solaris9
                                                                 Recommended
  Mount Point                Device        FS Type    Size (MB)  Min Size(MB)
  /                                        ufs                   1025
  -                          c0t0d0s1      swap       257        3







 Esc     F2     F3     F4     F5     F6      F7      F8      F9   ^D   ^X
 HELP  CHOICE  SAVE  SLICE  PRINT  CANCEL SCHEDULE SPLIT  MERGE  CLR OTHR
```

FIGURE 38–5 Solaris Live Upgrade Configuration Menu

4. From the Configuration menu, press F2 to display the Choices menu.
5. Choose slice 0 from disk c0t15d0 for root (/).
6. From the configuration menu, create a new slice for swap on c0t15d0 by selecting a swap slice to be split.
7. Press F2 to display the Choices menu.
8. Select slice 1 from disk c0t15d0 for the new swap slice.
9. Press F3 to create the new boot environment.

Upgrade the Inactive Boot Environment

The operating system image is taken from the network for the upgrade.

1. From the main menu, select Upgrade.

```
Name of New Boot Environment:   c0t15d0s0
Package Media: /net/ins3-svr/export/Solaris_9/combined.solaris_wos
```

2. Press F3.

Activate the Inactive Boot Environment

The c0t15d0s0 boot environment is made bootable. The system is then rebooted and c0t15d0s0 becomes the active boot environment. The c0t4d0s0 boot environment is now inactive.

1. From the main menu, select Activate.

   ```
   Name of Boot Environment: c0t15d0s0
   Do you want to force a Live Upgrade sync operations: no
   ```

2. Press F3.
3. Press Return.
4. Type:

   ```
   # init 6
   ```

If a fallback is necessary, use the command-line procedures in the previous example: "Fall Back to the Source Boot Environment" on page 489.

CHAPTER 39

Solaris Live Upgrade (Command Reference)

The following list shows commands that you can type at the command line rather than by using the menus. The Solaris Live Upgrade includes man pages for all the listed command-line utilities.

TABLE 39–1 Solaris Live Upgrade Command-Line Options

Task	Command
Activate an inactive boot environment.	`luactivate`(1M)
Cancel a scheduled copy or create job.	`lucancel`(1M)
Compare an active boot environment with an inactive boot environment.	`lucompare`(1M)
Recopy file systems to update an inactive boot environment.	`lucopy`
Create a boot environment.	`lucreate`(1M)
Name the active boot environment.	`lucurr`(1M)
Delete a boot environment.	`ludelete`(1M)
Add a description to a boot environment name.	`ludesc`(1M)
List critical file systems for each boot environment.	`lufslist`(1M)
Enable a mount of all of the file systems in a boot environment. This command enables you to modify the files in a boot environment while that boot environment is inactive.	`lumount`(1M)
Rename a boot environment.	`lurename`(1M)
List status of all boot environments.	`lustatus`(1M)

TABLE 39–1 Solaris Live Upgrade Command-Line Options *(Continued)*

Task	Command
Enable an unmount of all the file systems in a boot environment. This command enables you to modify the files in a boot environment while that boot environment is inactive.	`luumount`(1M)
Upgrade an operating environment or install a flash archive on an inactive boot environment.	`luupgrade`(1M)

CHAPTER 40

Installing Over a Wide Area Network With WAN Boot (Topics)

This section provides instructions for installing systems over a wide area network (WAN).

Chapter 41	Provides overview information on the WAN boot installation process.
Chapter 42	Provides information that you need to know before performing a WAN boot installation.
Chapter 43	Provides instructions about how to prepare for a WAN boot installation.
Chapter 44	Provides instructions for installing a client over a WAN.
Chapter 45	Provides examples of a WAN boot installation.
Chapter 46	Lists WAN boot commands and configuration file keywords and syntax.

CHAPTER **41**

WAN Boot (Overview)

This chapter provides an overview of the WAN boot installation method. This chapter describes the following topics.

What Is WAN Boot?

The WAN boot installation method enables you to boot and install software over a wide area network (WAN) by using HTTP. By using WAN boot, you can install the Solaris operating environment on SPARC based systems over a large public network where the network infrastructure might be untrustworthy. You can use WAN boot with security features to protect data confidentiality and installation image integrity.

The WAN boot installation method enables you to transmit an encrypted Solaris Flash archive over a public network to a remote SPARC based client. The WAN boot programs then install the client system by performing a custom JumpStart installation. To protect the integrity of the installation, you can use private keys to authenticate and encrypt data. You can also transmit your installation data and files over a secure HTTP connection by configuring your systems to use digital certificates.

To perform a WAN boot installation, you install a SPARC based system by downloading the following information from a web server over a HTTP or secure HTTP connection.

- `wanboot` program – The `wanboot` program is the second level boot program that loads the WAN boot miniroot, client configuration files, and installation files. The `wanboot` program performs tasks similar to those that are performed by the `ufsboot` or `inetboot` second level boot programs.

- WAN boot file system – WAN boot uses several different files to configure the client and retrieve data to install the client system. These files are located in the `/etc/netboot` directory of the web server. The `wanboot-cgi` program transmits thes files to the client as a file system, called the WAN boot file system.
- WAN boot miniroot – The WAN boot miniroot is a version of the Solaris miniroot that has been modified to perform a WAN boot installation. The WAN boot miniroot, like the Solaris miniroot, contains a kernel and just enough software to install the Solaris environment. The WAN boot miniroot contains a subset of the software in the Solaris miniroot.
- Custom JumpStart configuration files – To install the system, WAN boot transmits `sysidcfg`, `rules.ok`, and profile files to the client. WAN boot then uses these files to perform a custom JumpStart installation on the client system.
- Solaris Flash archive – A Solaris Flash archive is a collection of files that you copy from a master system. You can then use this archive to install a client system. WAN boot uses the custom JumpStart installation method to install a Solaris Flash archive on the client system. After you install an archive on a client system, the system contains the exact configuration of the master system.

You then install the archive on the client by using the custom JumpStart installation method.

You can protect the transfer of the previously listed information by using keys and digital certificates.

For a more detailed description of the sequence of events in a WAN boot installation, see "How WAN Boot Works (Overview)" on page 507.

When to Use WAN Boot

The WAN boot installation method enables you to install SPARC based systems that are located in geographically remote areas. You might want to use WAN boot to install remote servers or clients that are accessible only over a public network.

If you want to install systems that are located on your local area network (LAN), the WAN boot installation method might require more configuration and administration than necessary. For information about how to install systems over a LAN, see Chapter 12.

How WAN Boot Works (Overview)

WAN boot uses a combination of servers, configuration files, Common Gateway Interface (CGI) programs, and installation files to install a remote SPARC based client. This section describes the general sequence of events in a WAN boot installation.

Sequence of Events in a WAN Boot Installation

Figure 41–1 shows the basic sequence of events in a WAN boot installation. In this figure, a SPARC based client retrieves configuration data and installation files from a web server and an install server over a WAN.

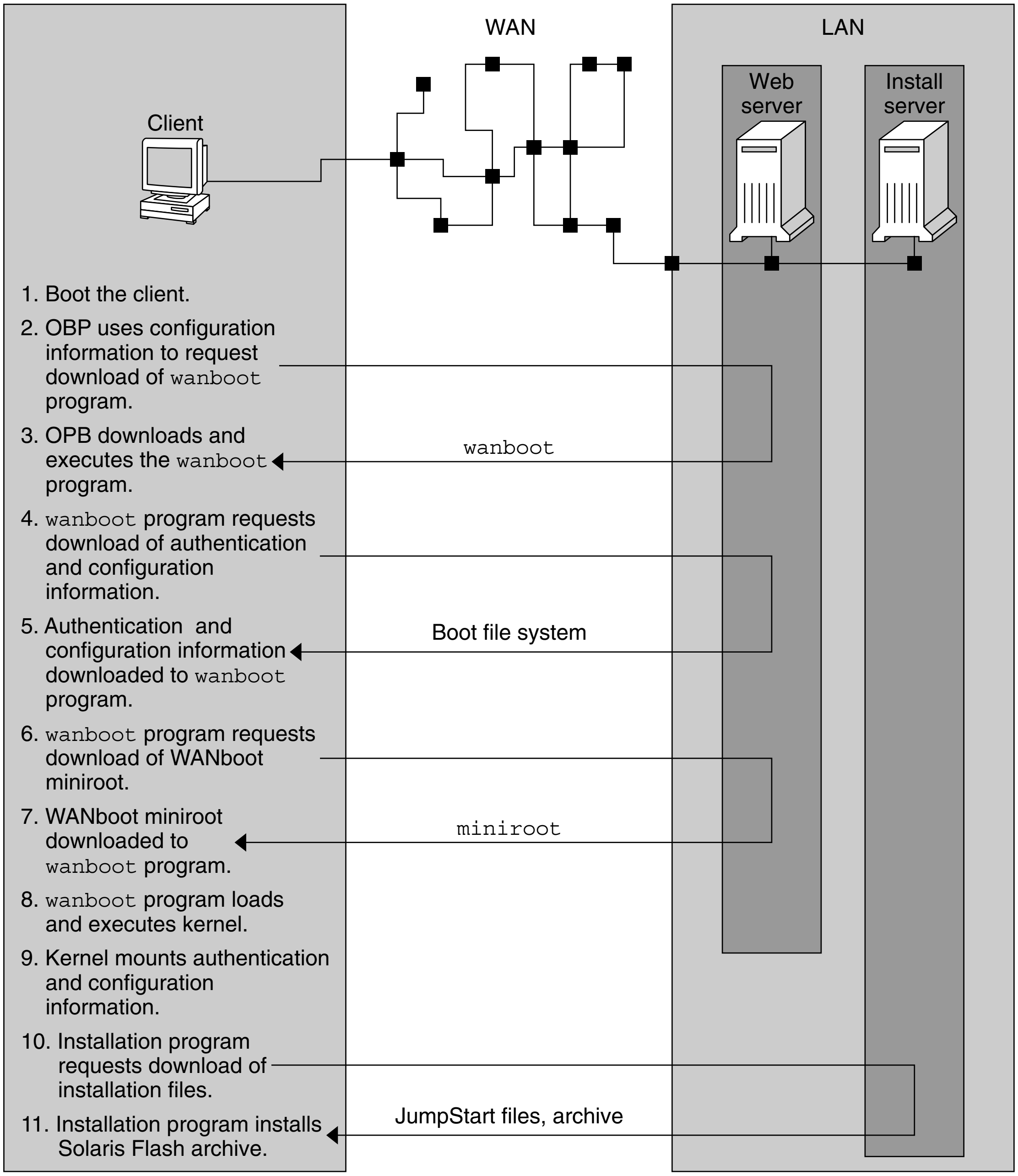

WAN
LAN
Client
Web server
Install server
1. Boot the client.
2. OBP uses configuration information to request download of wanboot program.
3. OPB downloads and executes the wanboot program.
wanboot
4. wanboot program requests download of authentication and configuration information.
5. Authentication and configuration information downloaded to wanboot program.
Boot file system
6. wanboot program requests download of WANboot miniroot.
7. WANboot miniroot downloaded to wanboot program.
miniroot
8. wanboot program loads and executes kernel.
9. Kernel mounts authentication and configuration information.
10. Installation program requests download of installation files.
11. Installation program installs Solaris Flash archive.
JumpStart files, archive

FIGURE 41–1 Sequence of Events in a WAN Boot Installation

1. You boot the client in one of the following ways.

- Boot from the network by setting network interface variables in the Open Boot PROM (OBP).
- Boot from the network with the DHCP option.
- Boot from a local CD-ROM.

2. The client OBP obtains configuration information from one of the following sources.
   - From boot argument values that are typed at the command line by the user
   - From the DHCP server, if the network uses DHCP
3. The client OBP requests the WAN boot second level boot program (`wanboot`).

   The client OBP downloads the `wanboot` program from the following sources.
   - From a special web server, called the WAN boot server, by using the Hyper Text Transfer Protocol (HTTP)
   - From a local CD-ROM (not shown in the figure)
4. The `wanboot` program requests the client configuration information from the WAN boot server.
5. The `wanboot` program downloads configuration files that are transmitted by the `wanboot-cgi` program from the WAN boot server. The configuration files are transmitted to the client as the WAN boot file system.
6. The `wanboot` program requests the download of the WAN boot miniroot from the WAN boot server.
7. The `wanboot` program downloads the WAN boot miniroot from the WAN boot server by using HTTP or secure HTTP.
8. The `wanboot` program loads and executes the UNIX kernel from the WAN boot miniroot.
9. The UNIX kernel locates and mounts the WAN boot file system for use by the Solaris installation program.
10. The installation program requests the download of a Solaris Flash archive and custom JumpStart files from an install server.

    The installation program downloads the archive and custom JumpStart files over an HTTP or HTTPS connection.
11. The installation program performs a custom JumpStart installation to install the Solaris Flash archive on the client.

Protecting Data During a WAN Boot Installation

The WAN boot installation method enables you to use hashing keys, encryption keys, and digital certificates to protect your system data during the installation. This section briefly describes the different data protection methods that are supported by the WAN boot installation method.

Checking the Integrity of Data With a Hashing Key

To protect the data you transmit from the WAN boot server to the client, you can generate a Hashed Message Authentication Code (HMAC) key. You install this hashing key on both the WAN boot server and the client. The WAN boot server uses this key to sign the data to be transmitted to the client. The client then uses this key to verify the integrity of the data that is transmitted by the WAN boot server. After you install a hashing key on a client, the client uses this key for future WAN boot installations.

For instructions about how to use a hashing key, see "Creating a Hashing Key and an Encryption Key" on page 541.

Encrypting Data With Encryption Keys

The WAN boot installation method enables you to encrypt the data you transmit from the WAN boot server to the client. You can use WAN boot utilities to create a Triple Data Encryption Standard (3DES) or Advanced Encryption Standard (AES) encryption key. You can then provide this key to both the WAN boot server and the client. WAN boot uses this encryption key to encrypt the data sent from the WAN boot server to the client. The client can then use this key to decrypt the encrypted configuration files and security files that are transmitted during the installation.

Once you install an encryption key on a client, the client uses this key for future WAN boot installations.

Your site might not permit the use of encryption keys. To determine if your site permits encryption, ask your site's security administrator. If your site permits encryption, ask your security administrator which type of encryption key, either 3DES or AES, you should use.

For instructions on how to use encryption keys, see "Creating a Hashing Key and an Encryption Key" on page 541.

Protecting Data by Using HTTPS

WAN boot supports the use of HTTP over Secure Sockets Layer (HTTPS) to transfer data between the WAN boot server and the client. By using HTTPS, you can require the server, or both the server and the client, to authenticate themselves during the installation. HTTPS also encrypts the data that is transferred from the server to the client during the installation.

HTTPS uses digital certificates to authenticate systems that exchange data over the network. A digital certificate is a file that identifies a system, either a server or client, as a system to trust during online communication. You can request a digital certificate from an external certificate authority, or create your own certificate and certificate authority.

To enable the client to trust the server and accept data from the server, you must install a digital certificate on the server. You then instruct the client to trust this certificate. You can also require the client to authenticate itself to the servers by providing a digital certificate to the client. You can then instruct the server to accept the certificate's signer when the client presents the certificate during the installation.

To use digital certificates during the installation, you must configure your web server to use HTTPS. See your web server documentation for information about how to use HTTPS.

For information about the requirements to use digital certificates during your WAN boot installation, see "Digital Certificate Requirements" on page 521. For instructions about how to use digital certificates in your WAN boot installation, see "Using Digital Certificates for Server and Client Authentication" on page 539.

Security Configurations Supported by WAN Boot (Overview)

WAN boot supports varying levels of security. You can use a combination of the security features that are supported in WAN boot to meet the needs of your network. A more secure configuration requires more administration, but also protects your system data to a greater extent. For more critical systems, or those systems you want to install over a public network, you might choose the configuration in "Secure WAN Boot Installation Configuration" on page 511. For less critical systems, or systems on semi-private networks, consider the configuration that is described in "Insecure WAN Boot Installation Configuration" on page 512.

This section briefly describes the different configurations you can use to set the level of security for your WAN boot installation. The section also describes the security mechanisms that are required by these configurations.

Secure WAN Boot Installation Configuration

This configuration protects the integrity of the data exchanged between the server and client, and helps keep the contents of the exchange confidential. This configuration uses an HTTPS connection, and uses either the 3DES or AES algorithm to encrypt the client configuration files. This configuration also requires the server to authenticate itself to the client during the installation. A secure WAN boot installation requires the following security features.

- HTTPS enabled on the WAN boot server and the install server
- HMAC SHA1 hashing key on the WAN boot server and the client

- 3DES or AES encryption key for the WAN boot server and the client
- Digital certificate of a certificate authority for the WAN boot server

If you want to also require client authentication during the installation, you must also use the following security features.

- Private key for the WAN boot server
- Digital certificate for the client

For a list of the tasks that are required to install with this configuration, see Table 43–1.

Insecure WAN Boot Installation Configuration

This security configuration requires the least administration effort, but provides the least secure transfer of data from the web server to the client. You do not need to create a hashing key, encryption key, or digital certificates. You do not need to configure your web server to use HTTPS. However, this configuration transfers the installation data and files over an HTTP connection, which leaves your installation vulnerable to interception over the network.

If you want the client to check the integrity of the data that is transmitted, you can use a HMAC SHA1 hashing key with this configuration. However, the Solaris Flash archive is not protected by the hashing key. The archive is transferred insecurely between the server and the client during the installation.

For a list of the tasks that are required to install with this configuration, see Table 43–2.

CHAPTER 42

Preparing to Install With WAN Boot (Planning)

This chapter describes how to prepare your network for a WAN boot installation. This chapter describes the following topics.

- "WAN Boot Requirements and Guidelines" on page 513
- "WAN Boot Security Limitations" on page 522
- "Gathering Information for WAN Boot Installations" on page 522

WAN Boot Requirements and Guidelines

The section describes the system requirements to perform a WAN boot installation.

TABLE 42–1 System Requirements for WAN Boot Installation

System and Description	Requirements
WAN boot server – The WAN boot server is a web server that provides the `wanboot` program, the configuration and security files, and the WAN boot miniroot.	■ Operating system – Solaris 9 12/03 operating environment, or compatible version ■ Must be configured as web server ■ Web server software must support HTTP 1.1 ■ If you want to use digital certificates, the web server software must support HTTPS

TABLE 42–1 System Requirements for WAN Boot Installation *(Continued)*

System and Description	Requirements
Install server – The install server provides the Solaris Flash archive and custom JumpStart files that are required to install the client.	■ Available disk space – space for each Solaris Flash archive ■ Media drive – CD-ROM or DVD-ROM drive ■ Operating system – Solaris 9 12/03 operating environment, or compatible version If the install server is a different system than the WAN boot server, the install server must meet these additional requirements. ■ Must be configured as a web server ■ Web server software must support HTTP 1.1 ■ If you want to use digital certificates, the web server software must support HTTPS
Client system – The remote system you want to install over a WAN	■ Memory - Minimum of 256 Mbytes of RAM ■ CPU – UltraSPARC II processor minimum ■ Hard disk – At least 2 Gbytes of hard disk space ■ OBP – WAN boot-enabled PROM If the client does not have the appropriate PROM, the client must have a CD-ROM drive. To determine if your client has a WAN boot-enabled PROM, see "To Check the Client OBP for WAN Boot Support" on page 559.
(Optional) DHCP server – You can use a DHCP server to provide client configuration information.	If you are using a SunOS DHCP server, you must perform one of the following tasks. ■ Upgrade the server to be an EDHCP server. ■ Rename Sun vendor options to satisfy the eight-character limit on options. For more information about the WAN installation-specific Sun vendor options, see "(Optional) Providing Configuration Information With a DHCP Server" on page 555. If the DHCP server is on a different subnet than the client, you must configure a BOOTP relay agent. For more information about how to configure a BOOTP relay agent, see "Configuring DHCP Service (Task)" in *System Administration Guide: IP Services*.
(Optional) Logging server – By default, all booting and installation log messages are displayed on the client console during a WAN installation. If you want to view these messages on another system, you can specify a system to serve as a logging server.	Must be configured as web server. **Note –** If you use HTTPS during your installation, the logging server must be the same system as the WAN boot server.

TABLE 42–1 System Requirements for WAN Boot Installation *(Continued)*

System and Description	Requirements
(Optional) Proxy server – You can configure the WAN boot feature to use an HTTP proxy during the download of the installation data and files.	If the installation uses HTTPS, the proxy server must be configured to tunnel HTTPS.

Web Server Software Requirements and Guidelines

The web server software you use on your WAN boot server and install server must meet the following requirements.

- Operating system requirements – WAN boot provides a Common Gateway Interface (CGI) program (`wanboot-cgi`) that converts data and files into a specific format that the client machine expects. To perform a WAN boot installation with these scripts, the web server software must run on the Solaris 9 12/03 operating environment, or compatible version.
- File size limitations – Your web server software might limit the size of the files you can transmit over HTTP. Check your web server documentation to make sure the software can transmit files that are the size of a Solaris Flash archive.
- SSL support – If you want to use HTTPS in your WAN boot installation, the web server software must support SSL version 3.

Server Configuration Options

You can customize the configuration of the servers that are required by WAN boot to meet your network needs. You can host all the servers on one system, or place the servers on multiple systems.

- **Single server** – If you want to centralize the WAN boot data and files on one system, you can host all the servers on the same machine. You can administer all your different servers on one system, and you only need to configure one system as a web server. However, a single server might not be able to support the volume of traffic that is required for a large number of simultaneous WAN boot installations.
- **Multiple servers** – If you want to distribute the installation data and files across your network, you can host these servers on multiple machines. You might set up a central WAN boot server, and configure multiple install servers to host Solaris Flash archives across your network. If you host the install server and logging server on independent machines, you must configure those servers as web servers.

Storing Installation and Configuration Files in the Document Root Directory

The `wanboot-cgi` program transmits the following files during a WAN boot installation.

- `wanboot` program
- WAN boot miniroot
- Custom JumpStart files
- Solaris Flash archive

To enable the `wanboot-cgi` program to transmit these files you must store these files in a directory that is accessible to the web server software. One way to make these files accessible is to place these files in the *document root* on your web server.

The document root, or primary document directory, is the directory on your web server where you store files you want to make available to clients. You can name and configure this directory in your web server software. See your web server documentation for more information about setting up the document root directory on your web server.

You might want to create different subdirectories of the document root directory to store your different installation and configuration files. For example, you might want to create specific subdirectories for each group of clients that you want to install. If you plan to install several different releases of the Solaris operating environment across your network, you might create subdirectories for each release.

Figure 42–1 shows a basic sample structure for a document root directory. In this example, the WAN boot server and install server are on the same machine. The server is running the Apache web server software.

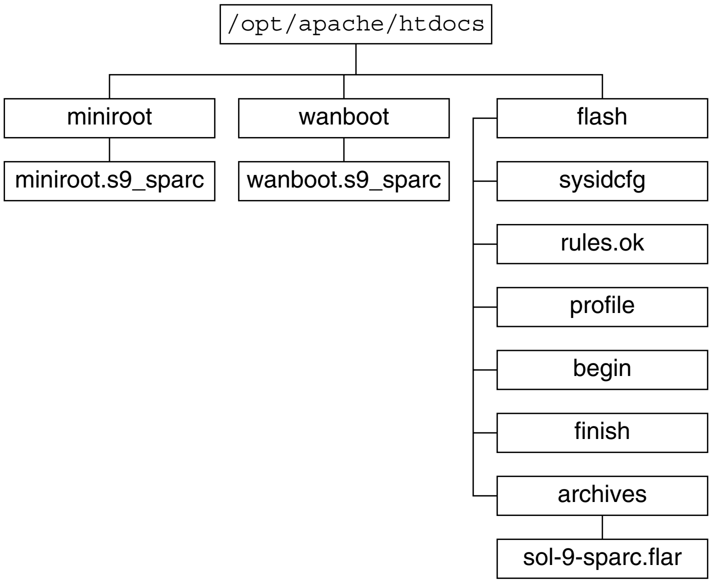


FIGURE 42–1 Sample Strucutre for Document Root Directory

This sample document directory uses the following structure.

- The /opt/apache/htdocs directory is the document root directory.
- The WAN boot miniroot (miniroot) directory contains the WAN boot miniroot.
- The wanboot directory contains the wanboot program.
- The Solaris Flash (flash) directory contains the custom JumpStart files that are required to install the client and the subdirectory archives. The archives directory contains the Solaris 9 Flash archive.

Note – If the WAN boot server and the install server are different systems, you might want to store the flash directory on the install server. Ensure that these files and directories are accessible to the WAN boot server.

For information about how to create the document root directory, see your web server documentation. For detailed instructions about how to create and store these installation files, see "Creating the Custom JumpStart Installation Files" on page 543.

Storing Configuration and Security Information in the /etc/netboot Hierarchy

The /etc/netboot directory contains the configuration information, private key, digital certificate, and certificate authority that are required for a WAN boot installation. This section describes the files and directories you can create in the /etc/netboot directory to customize your WAN boot installation.

Customizing the Scope of the WAN Boot Installation

During the installation, the wanboot-cgi program searches for the client information in the /etc/netboot directory on the WAN boot server. The wanboot-cgi program converts this information into the WAN boot file system, and then transmits the WAN boot file system to the client. You can create subdirectories within the /etc/netboot directory to customize the scope of the WAN installation. Use the following directory structures to define how configuration information is shared among the clients that you want to install.

- **Global configuration** – If you want all the clients on your network to share configuration information, store the files that you want to share in the /etc/netboot directory.
- **Network-specific configuration** – If you want only those machines on a specific subnet to share configuration information, store the configuration files that you want to share in a subdirectory of /etc/netboot. Have the subdirectory follow this naming convention.

  /etc/netboot/*net-ip*

  In this example, *net-ip* is the IP address of the client's subnet. For example, if you want all systems on the subnet with the IP address of 192.168.255.0 to share configuration files, create a /etc/netboot/192.168.255.0 directory. Then, store the configuration files in this directory.
- **Client-specific configuration** – If you want only a specific client to use the boot file system, store the boot file system files in a subdirectory of /etc/netboot. Have the subdirectory follow this naming convention.

  /etc/netboot/*net-ip*/*client-ID*

  In this example, *net-ip* is the IP address of the subnet. *client-ID* is either the client ID that is assigned by the DHCP server, or a user-specified client ID. For example, if you want a system with the client ID 010003BA152A42 on the subnet 192.168.255.0 to use specific configuration files, create a /etc/netboot/192.168.255.0/010003BA152A42 directory. Then, store the appropriate files in this directory.

Specifying Security and Configuration Information in the /etc/netboot Directory

You specify the security and configuration information by creating the following files and storing the files in the /etc/netboot directory.

- wanboot.conf – This file specifies the client configuration information for a WAN boot installation.
- System configuration file (system.conf) – This system configuration file specifies the location of the client's sysidcfg file and custom JumpStart files.
- keystore – This file contains the client's HMAC SHA1 hashing key, 3DES or AES encryption key, and SSL private key.
- truststore – This file contains the digital certificates of certificate signing authorities that the client should trust. These trusted certificates instruct the client to trust the server during the installation.
- certstore – This file contains the client's digital certificate.

Note – The certstore file must be located in the client ID directory. See "Customizing the Scope of the WAN Boot Installation" on page 518 for more information about subdirectories of the /etc/netboot directory.

For detailed instructions on how to create and store these files, see the following procedures.

- "Creating the System Configuration File" on page 549
- "Creating the wanboot.conf File" on page 551
- "Creating a Hashing Key and an Encryption Key" on page 541
- "Using Digital Certificates for Server and Client Authentication" on page 539

Sharing Security and Configuration Information in the /etc/netboot Directory

To install clients on your network, you might want to share security and configuration files among several different clients, or across entire subnets. You can share these files by distributing your configuration information throughout the /etc/netboot/*net-ip*/*client-ID*, /etc/netboot/*net-ip*, and /etc/netboot directories. The wanboot-cgi program searches these directories for the configuration information that best fits the client, and uses that information during the installation.

The wanboot—cgi program searches for client information in the following order.

1. /etc/netboot/*net-ip*/*client-ID* – The wanboot-cgi program first checks for configuration information that is specific to the client machine. If the /etc/netboot/*net-ip*/*client-ID* directory contains all the client configuration information, the wanboot-cgi program does not check for configuration

information elsewhere in the /etc/netboot directory.

2. /etc/netboot/*net-ip* – If all the required information is not located in the /etc/netboot/*net-ip*/*client-ID* directory, the wanboot-cgi program then checks for subnet configuration information in the /etc/netboot/*net-ip* directory.
3. /etc/netboot – If the remaining information is not located in the /etc/netboot/*net-ip* directory, the wanboot-cgi program then checks for global configuration information in the /etc/netboot directory.

Figure 42–2 demonstrates how you can set up the /etc/netboot directory to customize your WAN boot installations.

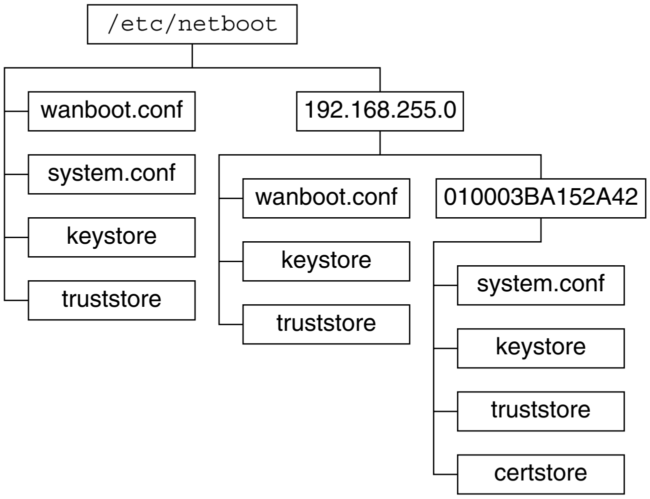


FIGURE 42–2 Sample /etc/netboot Directory

The /etc/netboot directory layout in Figure 42–2 enables you to perform the following WAN boot installations.

- When you install the client 010003BA152A42, the wanboot-cgi program uses the following files in the /etc/netboot/192.168.255.0/010003BA152A42 directory.
  - system.conf
  - keystore
  - truststore
  - certstore

  The wanboot-cgi program then uses the wanboot.conf file in the /etc/netboot/192.168.255.0 directory.
- When you install a client that is located on the 192.168.255.0 subnet, the wanboot-cgi program uses the wanboot.conf, keystore, and truststore files in the /etc/netboot/192.168.255.0 directory. The wanboot-cgi

program then uses the system.conf file in the /etc/netboot directory.

- When you install a client machine that is not located on the 192.168.255.0 subnet, the wanboot-cgi program uses the following files in the /etc/netboot directory.
  - wanboot.conf
  - system.conf
  - keystore
  - truststore

Storing the wanboot-cgi Program

The wanboot-cgi program transmits the data and files from the WAN boot server to the client. You must ensure that this program is in a directory on the WAN boot server that is accessible to the client. One method to make this program accessible to the client is to store this program in the cgi-bin directory of the WAN boot server. You might need to configure your web server software to use the wanboot-cgi program as a CGI program. See your web server documentation for information about CGI program requirements.

Digital Certificate Requirements

If you want to add security to your WAN boot installation, you can use digital certificates to enable server and the client authentication. WAN boot can use a digital certificate to establish the identity of the server or the client during an online transaction. Digital certificates are issued by a certificate authority (CA). These certificates contain a serial number, expiration dates, a copy of the certificate holder's public key, and the certificate authority's digital signature.

If you want to require server or both client and server authentication during your installation, you must install digital certificates on the server. Follow these guidelines when you use digital certificates.

- If you want to use digital certificates, the digital certificates must be formatted as part of a Public-Key Cryptography Standards #12 (PKCS#12) file.
- If you create your own certificates, you must create the certificates as PKCS#12 files.
- If you receive your certificates from third-party certificate authorities, request your certificates in the PKCS#12 format.

For detailed instructions on how to use PKCS#12 certificates during your WAN boot installation, see "Using Digital Certificates for Server and Client Authentication" on page 539.

WAN Boot Security Limitations

While WAN boot provides several different security features, WAN boot does not address these potential insecurities.

- **Denial of service (DoS) attacks** – A denial of service attack can take many forms, with the goal of preventing users from accessing a specific service. A DoS attack can overwhelm a network with large amounts of data, or aggressively consume limited resources. Other DoS attacks manipulate the data that is transmitted between systems in transit. The WAN boot installation method does not protect servers or clients from DoS attacks.
- **Corrupted binaries on the servers** – The WAN boot installation method does not check the integrity of the WAN boot miniroot or the Solaris Flash archive before you perform your installation. Before you perform your installation, check the integrity of your Solaris binaries against the Solaris Fingerprint Database at `http://sunsolve.sun.com`.
- **Encryption key and hashing key privacy** – If you use encryption keys or a hashing key with WAN boot, you must type the key value on the command line during your installation. Follow the precautions that are necessary for your network to make sure that these key values remain private.
- **Compromise of the network name service** – If you use a name service on your network, check the integrity of your name servers before you perform your WAN boot installation.

Gathering Information for WAN Boot Installations

You need to gather a wide variety of information to configure your network for a WAN boot installation. You might want to write down this information as you prepare to install over a WAN.

Use the following worksheets to record the WAN boot installation information for your network.

- Table 42–2
- Table 42–3

TABLE 42–2 Worksheet for Collecting Server Information

Information Needed	Notes
Install server information ■ Path to the WAN boot miniroot on install server ■ Path to the custom JumpStart files on the install server	
WAN boot server information ■ Path to the wanboot program on the WAN boot server ■ URL of the wanboot-cgi program on the WAN boot server ■ Path to the client's subdirectory in the /etc/netboot hierarchy on the WAN boot server ■ (Optional) File name of the PKCS#12 certificate file ■ (Optional) Host names of any machines other than the WAN boot server that are required for WAN installation ■ (Optional) IP address and TCP port number of the network's proxy server	
Optional server information ■ URL of the bootlog-cgi script on logging server ■ IP address and TCP port number of the network's proxy server	

TABLE 42–3 Worksheet for Collecting Client Information

Information	Notes
IP address for the client's subnet	
IP address for the client's router	
IP address of the client	
Subnet mask for the client	
Host name for the client	
MAC address of the client	

CHAPTER 43

Preparing to Install With WAN Boot (Tasks)

This chapter describes the following tasks that are necessary to prepare your network for a WAN boot installation.

- "Preparing to Install Over a Wide Area Network (Task Maps)" on page 525
- "Configuring the WAN Boot Server" on page 529
- "Creating the Custom JumpStart Installation Files" on page 543
- "Creating the Configuration Files" on page 549
- "(Optional) Providing Configuration Information With a DHCP Server" on page 555
- "(Optional) Configuring the WAN Boot Logging Server" on page 537

Preparing to Install Over a Wide Area Network (Task Maps)

The following tables list the tasks you need to perform to prepare for a WAN boot installation.

- For a list of the tasks you need to perform to prepare for a secure WAN boot installation, see Table 43–1.

  For a description of a secure WAN boot installation over HTTPS, see "Secure WAN Boot Installation Configuration" on page 511.
- For a list of the tasks you need to perform to prepare for an insecure WAN boot installation, see Table 43–2.

  For a description of an insecure WAN boot installation, see "Insecure WAN Boot Installation Configuration" on page 512.

To use a DHCP server or a logging server, complete the optional tasks that are listed at the bottom of each table.

TABLE 43–1 Task Map: Preparing to Perform a Secure WAN Boot Installation

Task	Description	For Instructions
Decide what security features you want to use in your installation.	Review the security features and configurations to decide what level of security you want to use in your WAN boot installation.	"Protecting Data During a WAN Boot Installation" on page 509 "Security Configurations Supported by WAN Boot (Overview)" on page 511
Collect WAN boot installation information.	Complete the worksheet to record all the information you need to perform a WAN boot installation.	"Gathering Information for WAN Boot Installations" on page 522
Create the document root directory on the WAN boot server.	Create the document root directory and any subdirectories to serve the configuration and installation files.	"Creating the Document Root Directory" on page 530
Create the WAN boot miniroot.	Use the `setup_install_server` command to create the WAN boot miniroot.	"SPARC: To Create a WAN Boot Miniroot" on page 530
Install the `wanboot` program on the WAN boot server.	Copy the `wanboot` program to the document root directory of the WAN boot server.	"Installing the `wanboot` Program on the WAN Boot Server" on page 532
Install the `wanboot-cgi` program on the WAN boot server.	Copy the `wanboot-cgi` program to the WAN boot server's CGI directory.	"To Copy the `wanboot-cgi` Program to the WAN Boot Server" on page 536
(Optional) Set up the logging server.	Configure a dedicated system for displaying boot and installation log messages.	"(Optional) Configuring the WAN Boot Logging Server" on page 537
Set up the `/etc/netboot` hierarchy.	Populate the `/etc/netboot` hierarchy with the configuration and security files that are required for a WAN boot installation.	"Creating the `/etc/netboot` Hierarchy on the WAN Boot Server" on page 534
Configure the web server to use secure HTTP for a more secure WAN boot installation.	Identify the web server requirements that are necessary to perform a WAN installation with HTTPS.	"(Optional) Protecting Data by Using HTTPS" on page 538

TABLE 43–1 Task Map: Preparing to Perform a Secure WAN Boot Installation *(Continued)*

Task	Description	For Instructions
Format digital certificates for a more secure WAN boot installation.	Split PKCS#12 file into a private key and a certificate to use with the WAN installation.	"Creating a Trusted Certificate and Client Private Key" on page 539
Create a hashing key and an encryption key for a more secure WAN boot installation.	Use the `wanbootutil keygen` command to create HMAC SHA1, 3DES, or AES keys.	"To Create a Hashing Key and Encryption Key" on page 541
Create the Solaris Flash archive.	Use the `flar create` command to create an archive of the software that you want to install on the client.	"To Create a Solaris Flash Archive" on page 544
Create the installation files for the custom JumpStart installation.	Use a text editor to create the following files: ■ `sysidcfg` ■ profile ■ `rules.ok` ■ begin scripts ■ finish scripts	"Creating the `sysidcfg` File" on page 545 "Creating the Profile" on page 546 "Creating the `rules` File" on page 547 "(Optional) Creating Begin and Finish Scripts" on page 549
Create the system configuration file.	Set the configuration information in the `system.conf` file.	"To Create a System Configuration File " on page 550
Create the WAN boot configuration file.	Set the configuration information in the `wanboot.conf` file.	"To Create a `wanboot.conf` File" on page 552
(Optional) Configure the DHCP server to support a WAN boot installation.	Set Sun vendor options and macros in the DHCP server.	"Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85

TABLE 43–2 Task Map: Preparing to Perform an Insecure WAN Boot Installation

Task	Description	For Instructions
Decide what security features you want to use in your installation.	Review the security features and configurations to decide what level of security you want to use in your WAN boot installation.	"Protecting Data During a WAN Boot Installation" on page 509 "Security Configurations Supported by WAN Boot (Overview)" on page 511

TABLE 43–2 Task Map: Preparing to Perform an Insecure WAN Boot Installation *(Continued)*

Task	Description	For Instructions
Collect WAN boot installation information.	Complete the worksheet to record all the information you need to perform a WAN boot installation.	“Gathering Information for WAN Boot Installations” on page 522
Create the document root directory on the WAN boot server.	Create the document root directory and any subdirectories to serve the configuration and installation files.	“Creating the Document Root Directory” on page 530
Create the WAN boot miniroot.	Use the `setup_install_server` command to create the WAN boot miniroot.	“SPARC: To Create a WAN Boot Miniroot” on page 530
Install the `wanboot` program on the WAN boot server.	Copy the `wanboot` program to the document root directory of the WAN boot server.	“Installing the `wanboot` Program on the WAN Boot Server” on page 532
Install the `wanboot-cgi` program on the WAN boot server.	Copy the `wanboot-cgi` program to the WAN boot server’s CGI directory.	“To Copy the `wanboot-cgi` Program to the WAN Boot Server” on page 536
(Optional) Set up the logging server.	Configure a dedicated system for displaying boot and installation log messages.	“(Optional) Configuring the WAN Boot Logging Server” on page 537
Set up the `/etc/netboot` hierarchy.	Populate the `/etc/netboot` hierarchy with the configuration and security files that are required for a WAN boot installation.	“Creating the `/etc/netboot` Hierarchy on the WAN Boot Server” on page 534
(Optional) Create a hashing key.	Use the `wanbootutil keygen` command to create HMAC SHA1 key. For insecure installations that check data integrity, complete this task to create an HMAC SHA1 hashing key.	“To Create a Hashing Key and Encryption Key” on page 541
Create the Solaris Flash archive.	Use the `flar create` command to create an archive of the software that you want to install on the client.	“To Create a Solaris Flash Archive” on page 544

TABLE 43–2 Task Map: Preparing to Perform an Insecure WAN Boot Installation *(Continued)*

Task	Description	For Instructions
Create the installation files for the custom JumpStart installation.	Use a text editor to create the following files: ■ sysidcfg ■ profile ■ rules.ok ■ begin scripts ■ finish scripts	"Creating the sysidcfg File" on page 545 "Creating the Profile" on page 546 "Creating the rules File" on page 547 "(Optional) Creating Begin and Finish Scripts" on page 549
Create the system configuration file.	Set the configuration information in the system.conf file.	"To Create a System Configuration File " on page 550
Create the WAN boot configuration file.	Set the configuration information in the wanboot.conf file.	"To Create a wanboot.conf File" on page 552
(Optional) Configure the DHCP server to support a WAN boot installation.	Set Sun vendor options and macros in the DHCP server.	"Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85

Configuring the WAN Boot Server

The WAN boot server is a web server that provides the boot and configuration data during a WAN boot installation. For a list of the system requirements for the WAN boot server, see Table 42–1.

This section describes the following tasks required to configure the WAN boot server for a WAN boot installation.

- "Creating the Document Root Directory" on page 530
- "Creating the WAN Boot Miniroot" on page 530
- "Installing the wanboot Program on the WAN Boot Server" on page 532
- "Creating the /etc/netboot Hierarchy on the WAN Boot Server" on page 534
- "Copying the WAN Boot CGI Program to the WAN Boot Server" on page 536
- "(Optional) Protecting Data by Using HTTPS" on page 538

Creating the Document Root Directory

To serve the configuration and installation files, you must make these files accessible to the web server software on the WAN boot server. One method to make these files accessible is to store them in the WAN boot server's document root directory.

If you want to use a document root directory to serve the configuration and installation files, you must create this directory. See your web server documentation for information about how to create the document root directory. For detailed information about how to design your document root directory, see "Storing Installation and Configuration Files in the Document Root Directory" on page 516.

Creating the WAN Boot Miniroot

WAN boot uses a special Solaris miniroot that has been modified to perform a WAN boot installation. The WAN boot miniroot contains a subset of the software in the Solaris miniroot. To perform a WAN boot installation, you must copy the miniroot from the Solaris DVD or the Solaris Software 1 of 2 CD to the WAN boot server. Use the `-w` option to the `setup_install_server` command to copy the WAN boot miniroot from the Solaris software media to your system's hard disk.

This procedure creates a SPARC WAN boot miniroot with SPARC media. If you want to serve a SPARC WAN boot miniroot from an x86–based server, you must create the miniroot on a SPARC machine. After you create the miniroot, copy the miniroot to the document root directory on the x86–based server.

For additional information about the `setup_install_server` command, see Chapter 15.

▼ SPARC: To Create a WAN Boot Miniroot

This procedure assumes that the WAN boot server is running the Volume Manager. If you are not using the Volume Manager, see *System Administration Guide: Devices and File Systems* for information about managing removable media without the Volume Manager.

1. **Become superuser on the WAN boot server.**

   The system must meet the following requirements.

   - Include a CD-ROM or DVD-ROM drive
   - Be part of the site's network and name service.

     If you use a name service, the system must already be in a name service, such as NIS, NIS+, DNS, or LDAP. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Insert the Solaris Software 1 of 2 CD or the Solaris DVD in the install server's drive.**

3. **Create a directory for the WAN boot miniroot and Solaris installation image.**

   ```
   # mkdir -p wan-dir-path  install-dir-path
   ```

   `-p` Instructs the `mkdir` command to create all the necessary parent directories for the directory you want to create.

   wan-dir-path Specifies the directory where the WAN boot miniroot is to be created on the install server. This directory needs to accommodate miniroots that are typically 250 Mbytes in size.

   install-dir-path Specifies the directory on the install server where the Solaris software image is to be copied. This directory can be removed later in this procedure.

4. **Change to the `Tools` directory on the mounted disc.**

   ```
   # cd /cdrom/cdrom0/s0/Solaris_9/Tools
   ```

   In the previous example, **`cdrom0`** is the path to the drive that contains the Solaris operating environment media.

5. **Copy the WAN boot miniroot and the Solaris software image to the WAN boot server's hard disk.**

   ```
   # ./setup_install_server -w wan-dir-path  install-dir-path
   ```

   wan-dir-path Specifies the directory where the WAN boot miniroot is to be copied

   install-dir-path Specifies the directory where the Solaris software image is to be copied

   Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

   The `setup_install_server -w` command creates the WAN boot miniroot and a network installation image of the Solaris software.

6. **(Optional) Remove the network installation image.**

   You do not need the Solaris software image to perform a WAN installation with a Solaris Flash archive. You can free up disk space if you do not plan to use the network installation image for other network installations. Type the following command to remove the network installation image.

   ```
   # rm -rf install-dir-path
   ```

7. **Make the WAN boot miniroot available to the WAN boot server in one of the following ways.**

- Create a symbolic link to the WAN boot miniroot in the document root directory of the WAN boot server.

  ```
  # cd /document-root-directory/miniroot
  # ln -s /wan-dir-path/miniroot .
  ```

  document-root-directory/`miniroot`

  Specifies the directory in the WAN boot server's document root directory where you want to link to the WAN boot miniroot.

  /*wan-dir-path*/`miniroot`

  Specifies the path to the WAN boot miniroot.

- Move the WAN boot miniroot to the document root directory on the WAN boot server.

  ```
  # mv /wan-dir-path/miniroot /document-root-directory/miniroot/miniroot-name
  ```

  wan-dir-path/`miniroot`

  Specifies the path to the WAN boot miniroot.

  /*document-root-directory*/`miniroot/`

  Specifies the path to the WAN boot miniroot directory in the WAN boot server's document root directory.

  miniroot-name

  Specifies the name of the WAN boot miniroot. Name the file descriptively, for example `miniroot.s9_sparc`.

Installing the `wanboot` Program on the WAN Boot Server

WAN boot uses a special second-level boot program (`wanboot`) to install the client. The `wanboot` program loads the WAN boot miniroot, client configuration files, and installation files that are required to perform a WAN boot installation.

To perform a WAN boot installation, you must provide the `wanboot` program to the client during the installation. You can provide this program to the client in the following ways.

- If your client's PROM supports WAN boot, you can transmit the program from the WAN boot server to the client. To check if your client's PROM supports WAN boot, see "Checking the Client OBP for WAN Boot Support" on page 559.
- If your client's PROM does not support WAN boot, you must provide the program to the client on a local CD. If your client's PROM does not support WAN boot, go to "Creating the `/etc/netboot` Hierarchy on the WAN Boot Server" on page 534 to continue preparing for your installation.

▼ SPARC: To Install the `wanboot` Program on the WAN Boot Server

This procedure assumes that the WAN boot server is running the Volume Manager. If you are not using the Volume Manager, see *System Administration Guide: Devices and File Systems* for information about managing removable media without the Volume Manager.

1. **Become superuser on the install server.**

2. **Insert the Solaris Software 1 of 2 CD or the Solaris DVD in the install server's drive.**

3. **Change to the `sun4u` platform directory on the Solaris Software 1 of 2 CD or the Solaris DVD.**

   ```
   # cd /cdrom/cdrom0/s0/Solaris_9/Tools/Boot/platform/sun4u/
   ```

4. **Copy the `wanboot` program to the install server.**

   ```
   # cp wanboot /document-root-directory/wanboot/wanboot-name
   ```

   document-root-directory Specifies the document root directory of the WAN boot server.

   wanboot-name Specifies the name of the `wanboot` program. Name this file descriptively, for example, `wanboot.s9_sparc`.

5. **Make the `wanboot` program available to the WAN boot server in one of the following ways.**

   - Create a symbolic link to the `wanboot` program in the document root directory of the WAN boot server.

     ```
     # cd /document-root-directory/wanboot
     # ln -s /wan-dir-path/wanboot .
     ```

     document-root-directory/`wanboot` Specifies the directory in the WAN boot server's document root directory where you want to link to the `wanboot` program.

     /*wan-dir-path*/`wanboot` Specifies the path to the `wanboot` program.

   - Move the WAN boot miniroot to the document root directory on the WAN boot server.

     ```
     # mv /wan-dir-path/wanboot /document-root-directory/wanboot/wanboot-name
     ```

     wan-dir-path/`wanboot` Specifies the path to the `wanboot` program.

/document-root-directory/wanboot/ — Specifies the path to the wanboot program directory in the WAN boot server's document root directory.

wanboot-name — Specifies the name of the wanboot program. Name the file descriptively, for example wanboot.s9_sparc.

Creating the /etc/netboot Hierarchy on the WAN Boot Server

During the installation, WAN boot refers to the contents of the /etc/netboot hierarchy on the web server for instructions about how to perform the installation. This directory contains the configuration information, private key, digital certificate, and certificate authority required for a WAN boot installation. During the installation, the wanboot-cgi program converts this information into the WAN boot file system. The wanboot-cgi program then transmits the WAN boot file system to the client.

You can create subdirectories within the /etc/netboot directory to customize the scope of the WAN installation. Use the following directory structures to define how configuration information is shared among the clients that you want to install.

- **Global configuration** – If you want all the clients on your network to share configuration information, store the files that you want to share in the /etc/netboot directory.
- **Network-specific configuration** – If you want only those machines on a specific subnet to share configuration information, store the configuration files that you want to share in a subdirectory of /etc/netboot. Have the subdirectory follow this naming convention.

  ```
  /etc/netboot/net-ip
  ```

  In this example, *net-ip* is the IP address of the client's subnet.
- **Client-specific configuration** – If you want only a specific client to use the boot file system, store the boot file system files in a subdirectory of /etc/netboot. Have the subdirectory follow this naming convention.

  ```
  /etc/netboot/net-ip/client-ID
  ```

  In this example, *net-ip* is the IP address of the subnet. *client-ID* is either the client ID that is assigned by the DHCP server, or a user-specified client ID.

For detailed planning information about how to design the /etc/netboot hierarchy, see "Storing Configuration and Security Information in the /etc/netboot Hierarchy" on page 518.

▼ To Create the `/etc/netboot` Hierarchy

1. **Become superuser on the WAN boot server.**

2. **Create the `/etc/netboot` directory.**

   ```
   # mkdir /etc/netboot
   ```

3. **Change the permissions of the `/etc/netboot` directory to 700.**

   ```
   # chmod 700 /etc/netboot
   ```

4. **Change the owner of the `/etc/netboot` directory to the web server owner.**

   ```
   # chown web-server-user:web-server-group /etc/netboot/
   ```

   web-server-user Specifies the user owner of the web server process

   web-server-group Specifies the group owner of the web server process

5. **Exit the superuser role.**

   ```
   # exit
   ```

6. **Assume the user role of the web server owner.**

7. **Create the client subdirectory of the `/etc/netboot` directory.**

   ```
   # mkdir -p /etc/netboot/net-ip/client-ID
   ```

   `-p` Instructs the `mkdir` command to create all the necessary parent directories for the directory you want to create

   (Optional) *net-ip* Specifies the network IP address of the client's subnet.

   (Optional) *client-ID* Specifies the client ID. The client ID can be a user-defined value or the DHCP client ID. The *client-ID* directory must be a subdirectory of the *net-ip* directory.

8. **For each directory in the `/etc/netboot` hierarchy, change the permissions to 700.**

   ```
   # chmod 700 /etc/netboot/dir-name
   ```

   dir-name Specifies the name of a directory in the `/etc/netboot` hierarchy

EXAMPLE 43–1 Creating the `/etc/netboot` Hierarchy on the WAN Boot Server

The following example shows how to create the `/etc/netboot` hierarchy for the client 010003BA152A42 on subnet 192.168.255.0. In this example, the user `nobody` and the group `admin` own the web server process.

The commands in this example perform the following tasks.

- Create the `/etc/netboot` directory.

EXAMPLE 43–1 Creating the /etc/netboot Hierarchy on the WAN Boot Server *(Continued)*

- Change the permissions of the /etc/netboot directory to 700.
- Change the ownership of the /etc/netboot directory to the owner of the web server process.
- Assume the same user role as the web server user.
- Create a subdirectory of /etc/netboot that is named after the subnet (192.168.255.0).
- Create a subdirectory of the subnet directory that is named after the client ID.
- Change the permissions of the /etc/netboot subdirectories to 700.

```
# cd /
# mkdir /etc/netboot/
# chmod 700 /etc/netboot
# chown nobody:admin /etc/netboot
# exit
server# su nobody
Password:
nobody# mkdir -p /etc/netboot/192.168.255.0/010003BA152A42
nobody# chmod 700 /etc/netboot/192.168.255.0
nobody# chmod 700 /etc/netboot/192.168.255.0/010003BA152A42
```

Copying the WAN Boot CGI Program to the WAN Boot Server

The wanboot-cgi program creates the data streams that transmit the the following files from the WAN boot server to the client.

- wanboot program
- WAN boot file system
- WAN boot miniroot

The wanboot-cgi program is installed on the system when you install the Solaris 9 12/03 operating environment, or compatible version. To enable the WAN boot server to use this program, copy this program to the cgi-bin directory of the WAN boot server.

▼ To Copy the wanboot-cgi Program to the WAN Boot Server

1. **Become superuser on the WAN boot server.**

2. **Copy the wanboot-cgi program to the WAN boot server.**

```
# cp /usr/lib/inet/wanboot/wanboot-cgi /WAN-server-root/cgi-bin/wanboot-cgi
```

/WAN-server-root Specifies the root directory of the web server software on the WAN boot server

3. **On the WAN boot server, change the permissions of the CGI program to 755.**

```
# chmod 755 /WAN-server-root/cgi-bin/wanboot-cgi
```

(Optional) Configuring the WAN Boot Logging Server

If you want to record boot and installation logging messages on a system other than the client, you must set up a logging server. If you want to use a logging server with HTTPS during the installation, you must configure the WAN boot server as the logging server.

To configure the logging server, follow these steps.

▼ To Configure the Logging Server

1. **Copy the `bootlog-cgi` script to the logging server's CGI script directory.**

```
# cp /usr/lib/inet/wanboot/bootlog-cgi \
  log-server-root/cgi-bin
```

log-server-root/`cgi-bin` Specifies the `cgi-bin` directory in the logging server's web server directory

2. **Change the permissions of the `bootlog-cgi` script to 755.**

```
# chmod 755 log-server-root/cgi-bin/bootlog-cgi
```

3. **Set the value of the `boot_logger` parameter in the `wanboot.conf` file.**

   In the `wanboot.conf` file, specify the URL of the `bootlog-cgi` script on the logging server.

   For more information about setting parameters in the `wanboot.conf` file, see "Creating the `wanboot.conf` File" on page 551.

   During the installation, boot and installation log messages are recorded in the `/tmp` directory of the logging server. The log file is named `bootlog.`*hostname*, where *hostname* is the host name of the client.

EXAMPLE 43–2 Configuring a Logging Server for WAN Boot Installation Over HTTPS

The following example configures the WAN boot server as a logging server.

EXAMPLE 43–2 Configuring a Logging Server for WAN Boot Installation Over HTTPS *(Continued)*

```
# cp /usr/lib/inet/wanboot/bootlog-cgi /opt/apache/cgi-bin/
# chmod 755 /opt/apache/cgi-bin/bootlog-cgi
```

(Optional) Protecting Data by Using HTTPS

To protect your data during the transfer from the WAN boot server to the client, you can use HTTP over Secure Sockets Layer (HTTPS). To use the more secure installation configuration that is described in "Secure WAN Boot Installation Configuration" on page 511, you must enable your web server to use HTTPS.

To enable the web server software on the WAN boot server to use HTTPS, you must perform the following tasks.

- Activate Secure Sockets Layer (SSL) support in your web server software.

  The processes for enabling SSL support and client authentication vary by web server. This document does not describe how to enable these security features on your web server. For information about these features, see the following documentation.

  - For information about activating SSL on the SunONE and iPlanet web servers, see the Sun ONE and iPlanet documentation collections on `http://docs.sun.com`.
  - For information about activating SSL on the Apache web server, see the Apache Documentation Project at `http://httpd.apache.org/docs-project/`.
  - If you are using web server software that is not listed in the previous list, see your web server software documentation.
- Install digital certificates on the WAN boot server.

  For information about using digital certificates with WAN boot, see "Using Digital Certificates for Server and Client Authentication" on page 539.
- Provide a trusted certificate to the client.

  For instructions about how to create a trusted certificate, see "Using Digital Certificates for Server and Client Authentication" on page 539.
- Create a hashing key and an encryption key.

  For instructions about how to create keys, see "Creating a Hashing Key and an Encryption Key" on page 541.
- (Optional) Configure the web server software to support client authentication.

  For information about how to configure your web server to support client authentication, see your web server documentation.

Using Digital Certificates for Server and Client Authentication

The WAN boot installation method can use PKCS#12 files to perform an installation over HTTPS with server or both client and server authentication. For requirements and guidelines about using PKCS#12 files, see "Digital Certificate Requirements" on page 521.

To use a PKCS#12 file in a WAN boot installation, you perform the following tasks.

- Split the PKCS#12 file into separate SSL private key and trusted certificate files.
- Insert the trusted certificate in the client's `truststore` file in the `/etc/netboot` hierarchy. The trusted certificate instructs the client to trust the server.
- (Optional) Insert the contents of the SSL private key file in the client's `keystore` file in the `/etc/netboot` hierarchy.

The `wanbootutil` command provides options to perform the tasks in the previous list.

Before you split a PKCS#12 file, create the appropriate subdirectories of the `/etc/netboot` hierarchy on the WAN boot server.

- For overview information that describes the `/etc/netboot` hierarchy, see "Storing Configuration and Security Information in the `/etc/netboot` Hierarchy" on page 518.
- For instructions about how to create the `/etc/netboot` hierarchy, see "Creating the `/etc/netboot` Hierarchy on the WAN Boot Server" on page 534.

▼ *Creating a Trusted Certificate and Client Private Key*

1. **Assume the same user role as the web server user on the WAN boot server.**

2. **Extract the trusted certificate from the PKCS#12 file. Insert the certificate in the client's `truststore` file in the `/etc/netboot` hierarchy.**

```
# wanbootutil p12split -i p12cert \
  -t /etc/netboot/net-ip/client-ID/truststore
```

`p12split`
Option to `wanbootutil` command that splits a PKCS#12 file into separate private key and certificate files.

`-i` *p12cert*
Specifies the name of the PKCS#12 file to split.

`-t /etc/netboot/`*net-ip*`/`*client-ID*`/truststore`
Inserts the certificate in the client's `truststore` file. *net-ip* is the IP address of the client's subnet. *client-ID* can be a user-defined ID or the DHCP client ID.

3. **(Optional) Decide if you want to require client authentication.**
   - If yes, continue with the following steps.
   - If no, go to "Creating a Hashing Key and an Encryption Key" on page 541.

   a. Insert the client certificate in the client's `certstore`.

   ```
   # wanbootutil p12split -i p12cert -c \
     /etc/netboot/net-ip/client-ID/certstore -k keyfile
   ```

   `p12split`
   Option to `wanbootutil` command that splits a PKCS#12 file into separate private key and certificate files.

   `-i` *p12cert*
   Specifies the name of the PKCS#12 file to split.

   `-c /etc/netboot/`*net-ip*`/`*client-ID*`/certstore`
   Inserts the client's certificate in the client's `certstore`. *net-ip* is the IP address of the client's subnet. *client-ID* can be a user-defined ID or the DHCP client ID.

   `-k` *keyfile*
   Specifies the name of the client's SSL private key file to create from the split PKCS#12 file.

   b. Insert the private key in the client's `keystore`.

   ```
   # wanbootutil keymgmt -i -k keyfile \
      -s /etc/netboot/net-ip/client-ID/keystore -o type=rsa
   ```

   `keymgmt -i`
   Inserts an SSL private key in the client's `keystore`

   `-k` *keyfile*
   Specifies the name of the client's private key file that was created in the previous step

   `-s /etc/netboot/`*net-ip*`/`*client-ID*`/keystore`
   Specifies the path to the client's `keystore`

   `-o type=rsa`
   Specifies the key type as RSA

EXAMPLE 43–3 Creating a Trusted Certificate for Server Authentication

In the following example, you use a PKCS#12 file to install client 010003BA152A42 on subnet 192.168.255.0. This command sample extracts a certificate from a PKCS#12 file that is named `client.p12`. The command then places the contents of the trusted certificate in the client's `truststore` file.

Before you execute these commands, you must first assume the same user role as the web server user. In this example, the web server user role is `nobody`.

```
server# su nobody
Password:
```

```
nobody# wanbootutil p12split -i client.p12 \
   -t /etc/netboot/192.168.255.0/010003BA152A42/truststore
nobody# chmod 600 /etc/netboot/192.168.255.0/010003BA152A42/truststore
```

Creating a Hashing Key and an Encryption Key

If you want to use HTTPS to transmit your data, you must create a HMAC SHA1 hashing key and an encryption key. If you plan to install over a semi-private network, you might not want to encrypt the installation data. You can use a HMAC SHA1 hashing key to check the integrity of the `wanboot` program. For overview information on hashing keys and encryption keys, see "Protecting Data During a WAN Boot Installation" on page 509.

By using the `wanbootutil keygen` command, you can generate these keys and store them in the appropriate `/etc/netboot` directory.

▼ *To Create a Hashing Key and Encryption Key*

1. **Assume the same user role as the web server user on the WAN boot server.**

2. **Create the master HMAC SHA1 key.**

```
# wanbootutil keygen -m
```

`keygen -m` Creates the master HMAC SHA1 key for the WAN boot server

3. **Create the HMAC SHA1 hashing key for the client from the master key.**

```
# wanbootutil keygen -c -o [net=net-ip,{cid=client-ID,}]type=sha1
```

`-c` Creates the client's hashing key from the master key.

`-o` Indicates that additional options are included for the `wanbootutil keygen` command.

(Optional) `net=`*net-ip* Specifies the IP address for the client's subnet. If you do not use the `net` option, the key is stored in the `/etc/netboot/keystore` file, and can be used by all WAN boot clients.

(Optional) `cid=`*client-ID* Specifies the client ID. The client ID can be a user-defined ID or the DHCP client ID. The `cid` option must be preceded by a valid `net=` value. If you do not specify the `cid` option with the `net` option, the key is stored in the `/etc/netboot/`*net-ip*`/keystore` file. This key can be used by all WAN boot clients on the *net-ip* subnet.

`type=sha1` Instructs the `wanbootutil keygen` utility to create a HMAC SHA1 hashing key for the client.

4. **Decide if you need to create an encryption key for the client.**

   You need to create an encryption key to perform a WAN boot installation over HTTPS. Before the client establishes an HTTPS connection with the WAN boot server, the WAN boot server transmits encrypted data and information to the client. The encryption key enables the client to decrypt this information and use this information during the installation.

   - If you are performing a more secure WAN installation over HTTPS with server authentication, continue.
   - If you only want to check the integrity of the `wanboot` program, you do not need to create an encryption key. Go to Step 6.

5. **Create an encryption key for the client.**

   ```
   # wanbootutil keygen -c -o [net=net-ip,{cid=client-ID,}]type=key-type
   ```

   `-c` Creates the client's encryption key.

   `-o` Indicates that additional options are included for the `wanbootutil keygen` command.

   (Optional) `net=`*net-ip* Specifies the network IP address for the client. If you do not use the `net` option, the key is stored in the `/etc/netboot/keystore` file, and can be used by all WAN boot clients.

   (Optional) `cid=`*client-ID* Specifies the client ID. The client ID can be a user-defined ID, or the DHCP client ID. The `cid` option must be preceded by a valid `net=` value. If you do not specify the `cid` option with the `net` option, the key is stored in the `/etc/netboot/`*net-ip*`/keystore` file. This key can be used by all WAN boot clients on the *net-ip* subnet.

   `type=`*key-type* Instructs the `wanbootutil keygen` utility to create an encryption key for the client. *key-type* can have a value of `3des` or `aes`.

6. **Install the keys on the client system.**

   For instructions about how to install keys on the client, see "Installing Keys on the Client" on page 561.

EXAMPLE 43–4 Creating Required Keys for WAN Boot Installation Over HTTPS

The following example creates a master HMAC SHA1 key for the WAN boot server. This example also creates a HMAC SHA1 hashing key and 3DES encryption key for client 010003BA152A42 on subnet 192.168.255.0.

EXAMPLE 43–4 Creating Required Keys for WAN Boot Installation Over HTTPS *(Continued)*

Before you execute these commands, you must first assume the same user role as the web server user. In this example, the web server user role is nobody.

```
server# su nobody
Password:
nobody# wanbootutil keygen -m
nobody# wanbootutil keygen -c -o net=192.168.255.0,cid=010003BA152A42,type=sha1
nobody# wanbootutil keygen -c -o net=192.168.255.0,cid=010003BA152A42,type=3des
```

Creating the Custom JumpStart Installation Files

WAN boot performs a custom JumpStart installation to install a Solaris Flash archive on the client. The custom JumpStart installation method is a command–line interface that enables you to automatically install several systems, based on profiles that you create. The profiles define specific software installation requirements. You can also incorporate shell scripts to include preinstallation and postinstallation tasks. You choose which profile and scripts to use for installation or upgrade. The custom JumpStart installation method installs or upgrades the system, based on the profile and scripts that you select. Also, you can use a sysidcfg file to specify configuration information so that the custom JumpStart installation is completely free of manual intervention.

To prepare the custom JumpStart files for a WAN boot installation, complete the following tasks.

- "Creating the Solaris Flash Archive" on page 544
- "Creating the sysidcfg File" on page 545
- "Creating the rules File" on page 547
- "Creating the Profile" on page 546
- "(Optional) Creating Begin and Finish Scripts" on page 549

For detailed information on the custom JumpStart installation method, see Chapter 25.

Creating the Solaris Flash Archive

The Solaris Flash installation feature enables you to use a single reference installation of the Solaris operating environment on a system, which is called the master system. You can then create a Solaris Flash archive, which is a replica image of the master system. You can install the Solaris Flash archive on other systems in the network, creating clone systems.

This section describes how to create a Solaris Flash archive to use in your WAN boot installation. Before you create a Solaris Flash archive, you must first install the master system.

- For information about installing a master system, see "Installing the Master System" on page 220.
- For detailed information about Solaris Flash archives, see Chapter 21.

▼ To Create a Solaris Flash Archive

For detailed instructions about how to create a Solaris Flash archive, see "Creating a Solaris Flash Archive" on page 226.

1. **Boot the master system.**

   Run the master system in as inactive a state as possible. When possible, run the system in single-user mode. If that is not possible, shut down any applications that you want to archive and any applications that require extensive operating system resources.

2. **To create the archive, use the `flar create` command.**

   ```
   # flar create -n name [optional-parameters]   document-root/flash/filename
   ```

   name — The name that you give the archive. The *name* you specify is the value of the `content_name` keyword.

   optional-parameters — You can use several options to the `flar create` command to customize your Solaris Flash archive. For detailed descriptions of these options, see Chapter 23.

   document-root/`flash` — The path to the Solaris Flash subdirectory of the install server's document root directory.

   filename — The name of the archive file.

   To conserve disk space, you might want to use the `-c` option to the `flar create` command to compress the archive. However, a compressed archive can affect the performance of your WAN boot installation. For more information about creating a compressed archive, see the man page `flar create`(1M).

   - If the archive creation is successful, the `flar create` command returns an exit code of 0.

- If the archive creation fails, the flar create command returns a nonzero exit code.

For examples of how to create a Solaris Flash archive, see “Examples—Creating an Archive for an Initial Install” on page 227.

Creating the sysidcfg File

You can specify a set of keywords in the sysidcfg file to preconfigure a system. For more detailed information about sysidcfg keywords and values, see “Preconfiguring With the sysidcfg File” on page 65.

▼ To Create the sysidcfg File

1. **Create a file called sysidcfg in a text editor on the install server.**
2. **Type the sysidcfg keywords you want.**

   For detailed information about sysidcfg keywords, see “sysidcfg File Keywords” on page 67.
3. **Save the sysidcfg file in a location that is accessible to the WAN boot server.**

   Save the file to one of the following locations.

   - If the WAN boot server and install server are hosted on the same machine, save this file to the flash subdirectory of the document root directory on the WAN boot server.
   - If the WAN boot server and install server are not on the same machine, save this file to the flash subdirectory of the document root directory of the install server.

EXAMPLE 43–5 sysidcfg File for WAN Boot Installation

The following is an example of a sysidcfg file for a SPARC based system. The host name, IP address, and netmask of this system have been preconfigured by editing the name service.

```
network_interface=primary {hostname=seahag
                            default_route=192.168.88.1
                            ip_address=192.168.88.210
                            netmask=255.255.0.0
                            protocol_ipv6=no}
timezone=US/Central
system_locale=C
terminal=xterm
timeserver=localhost
name_service=NIS {name_server=matter(192.168.255.255)
```

EXAMPLE 43–5 `sysidcfg` File for WAN Boot Installation *(Continued)*

```
                domain_name=mind.over.example.com
                }
security_policy=none
```

Creating the Profile

A profile is a text file that instructs the custom JumpStart program how to install the Solaris software on a system. A profile defines elements of the installation, for example, the software group to install.

For detailed information about how to create profiles, see "Creating a Profile" on page 277.

▼ To Create a Profile

1. **Create a text file on the install server. Name the file descriptively.**

   Ensure that the name of the profile reflects how you intend to use the profile to install the Solaris software on a system. For example, you might name the profiles `basic_install`, `eng_profile`, or `user_profile`.

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

   For a list of profile keywords and values, see "Profile Keywords and Values" on page 340.

   Profile keywords and their values are case sensitive.

3. **Save the profile in a location that is accessible to the WAN boot server.**

   Save the profile in one of the following locations.

   - If the WAN boot server and install server are hosted on the same machine, save this file to the `flash` subdirectory of the document root directory on the WAN boot server.
   - If the WAN boot server and install server are not on the same machine, save this file to the `flash` subdirectory of the document root directory of the install server.

4. **Ensure that `root` owns the profile and that the permissions are set to 644.**

5. **(Optional) Test the profile.**

   "Testing a Profile" on page 286 contains information about testing profiles.

EXAMPLE 43–6 Retrieving a Solaris Flash Archive From a Secure HTTP Server

In the following example, the profile indicates that the custom JumpStart program retrieves the Solaris Flash archive from a secure HTTP server.

```
# profile keywords         profile values
# ----------------         -------------------
install_type               flash_install
archive_location           https://192.168.255.255/solarisupdate.flar
partitioning               explicit
filesys                    c0t1d0s0 4000 /
filesys                    c0t1d0s1 512 swap
filesys                    c0t1d0s7 free /export/home
```

The following list describes some of the keywords and values from this example.

`install_type` The profile installs a Solaris Flash archive on the clone system. All files are overwritten as in an initial installation.

`archive_location` The compressed Solaris Flash archive is retrieved from a secure HTTP server.

`partitioning` The file system slices are determined by the `filesys` keywords, value `explicit`. The size of root (/) is based on the size of the Solaris Flash archive. The size of `swap` is set to the necessary size and is installed on `c0t1d0s1`. `/export/home` is based on the remaining disk space. `/export/home` is installed on `c0t1d0s7`.

Creating the `rules` File

The `rules` file is a text file that contains a rule for each group of systems on which you want to install the Solaris operating environment. Each rule distinguishes a group of systems that are based on one or more system attributes. Each rule also links each group to a profile. A profile is a text file that defines how the Solaris software is to be installed on each system in the group. For example, the following rule specifies that the JumpStart program use the information in the `basic_prof` profile to install any system with the `sun4u` platform group.

```
karch sun4u - basic_prof -
```

The `rules` file is used to create the `rules.ok` file, which is required for custom JumpStart installations.

For detailed information about how to create a `rules` file, see "Creating the `rules` File" on page 274.

▼ To Create a rules File

1. **On the install server, create a text file that is named rules.**

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

   For detailed information about how to create a rules file, see "Creating the rules File" on page 274.

3. **Save the rules file on the install server.**

4. **Validate the rules file.**

   ```
   $ ./check [-p path -r file-name]
   ```

   -p *path* Validates the rules by using the check script from the Solaris 9 software image instead of the check script from the system you are using. *path* is the image on a local disk or a mounted Solaris DVD or a Solaris Software 1 of 2 CD.

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

   -r *file_name* Specifies a rules file other than the file that is named rules. By using this option, you can test the validity of a rule before you integrate the rule into the rules file.

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

5. **Save the rules.ok file in a location that is accessible to the WAN boot server.**

   Save the file to one of the following locations.

   - If the WAN boot server and install server are hosted on the same machine, save this file to the flash subdirectory of the document root directory on the WAN boot server.
   - If the WAN boot server and install server are not on the same machine, save this file to the flash subdirectory of the document root directory of the install server.

6. **Ensure that root owns the rules.ok file and that the permissions are set to 644.**

For examples of rules files, see "rules File Example" on page 276.

(Optional) Creating Begin and Finish Scripts

Begin and finish scripts are user-defined Bourne shell scripts that you specify in the `rules` file. A begin script performs tasks before the Solaris software is installed on a system. A finish script performs tasks after the Solaris software is installed on a system, but before the system reboots. You can use these scripts only when using custom JumpStart to install Solaris.

You can use begin scripts to create derived profiles. Finish scripts enable you to perform various postinstallation tasks, such as adding files, packages, patches, or additional software.

You must store the begin and finish scripts in the same directory as the `sysidcfg`, `rules.ok`, and profile files on the install server.

- For more information about creating begin scripts, see "Creating Begin Scripts" on page 293.
- For more information about creating finish scripts, see "Creating Finish Scripts" on page 295.

Creating the Configuration Files

WAN boot uses the following files to specify the location of the data and files that are required for a WAN boot installation.

- system configuration file (`system.conf`)
- `wanboot.conf` file

This section describes how to create and store these two files.

Creating the System Configuration File

In the system configuration file, you can direct the WAN boot installation programs to the following files.

- `sysidcfg` file
- `rules.ok` file
- Custom JumpStart profile

WAN boot follows the pointers in the system configuration file to install and configure the client.

The system configuration file is a plain text file, and must be formatted in the following pattern.

setting=*value*

To use a system configuration file to direct the WAN installation programs to the `sysidcfg`, `rules.ok`, and profile files, follow these steps.

▼ To Create a System Configuration File

1. **Assume the same user role as the web server user on the WAN boot server.**

2. **Create a text file. Name the file descriptively, for example, `sys-conf.s9-sparc`.**

3. **Add the following entries to the system configuration file.**

   `SsysidCF=`*sysidcfg-file-URL*

   This setting points to the `flash` directory on the install server that contains the `sysidcfg` file. Make sure that this URL matches the path to the `sysidcfg` file that you created in "Creating the `sysidcfg` File" on page 545.

   For WAN installations that use HTTPS, set the value to a valid HTTPS URL.

   `SjumpsCF=`*jumpstart-files-URL*

   This setting points to the Solaris Flash directory on the install server that contains the `rules.ok`file, profile file, and begin and finish scripts. Make sure that this URL matches the path to the custom JumpStart files that you created in "Creating the Profile" on page 546 and "Creating the `rules` File" on page 547.

   For WAN installations that use HTTPS, set the value to a valid HTTPS URL.

4. **Save the file to a directory that is accessible to the WAN boot server.**

   For administration purposes, you might want to save the file to the appropriate client directory in the `/etc/netboot` directory on the WAN boot server.

5. **Change the permissions on the system configuration file to 600.**

   ```
   # chmod 600 /path/system-conf-file
   ```

   path — Specifies the path to the directory that contains the system configuration file.

   system-conf-file — Specifies the name of the system configuration file.

EXAMPLE 43–7 System Configuration File for WAN Boot Installation Over HTTPS

In the following example, the WAN boot programs check for the `sysidcfg` and custom JumpStart files on the web server `https://www.example.com` on port 1234. The web server uses secure HTTP to encrypt data and files during the installation.

EXAMPLE 43–7 System Configuration File for WAN Boot Installation Over HTTPS *(Continued)*

The `sysidcfg` and custom JumpStart files are located in the `flash` subdirectory of the document root directory `htdocs`.

```
SsysidCF=https://www.example.com:1234/htdocs/flash
SjumpsCF=https://www.example.com:1234/htdocs/flash
```

EXAMPLE 43–8 System Configuration File for Insecure WAN Boot Installation

In the following example, the WAN boot programs check for the `sysidcfg` and custom JumpStart files on the web server `http://www.example.com`. The web server uses HTTP, so the data and files are not protected during the installation.

The `sysidcfg` and custom JumpStart files are located in the `flash` subdirectory of the document root directory `htdocs`.

```
SsysidCF=http://www.example.com/htdocs/flash
SjumpsCF=http://www.example.com/htdocs/flash
```

Creating the `wanboot.conf` File

The `wanboot.conf` file is a plain text configuration file that the WAN boot programs use to perform a WAN installation. The `wanboot-cgi` program, the boot file system, and the WAN boot miniroot all use the information included in the `wanboot.conf` file to install the client machine.

Save the `wanboot.conf` file in the appropriate client subdirectory in the `/etc/netboot` hierarchy on the WAN boot server. For information about how to define the scope of your WAN boot installation with the `/etc/netboot` hierarchy, see "Creating the `/etc/netboot` Hierarchy on the WAN Boot Server" on page 534.

If the WAN boot server is running the Solaris 9 12/03 operating environment, or compatible version, a sample `wanboot.conf` file is located in `/etc/netboot/wanboot.conf.sample`. You can use this sample as a template for your WAN boot installation.

You must include the following information in the `wanboot.conf` file.

TABLE 43–3 Information for the `wanboot.conf` File

Type of Information	Description
WAN boot server information	■ Path to `wanboot` program on the WAN boot server ■ URL of `wanboot-cgi` program on WAN boot server

TABLE 43–3 Information for the wanboot.conf File *(Continued)*

Type of Information	Description
Install server information	■ Path to WAN boot miniroot on the install server ■ Path to system configuration file on the WAN boot server that specifies location of sysidcfg and custom JumpStart files
Security information	■ Signature type for the WAN boot file system or WAN boot miniroot ■ Encryption type for the WAN boot file system ■ Whether the server should be authenticated during the WAN boot installation ■ Whether the client should be authenticated during the WAN boot installation
Optional information	■ Additional hosts that might need to be resolved for the client during a WAN boot installation ■ URL to the bootlog-cgi script on the logging server

You specify this information by listing parameters with associated values in the following format.

parameter=value

For detailed information about wanboot.conf file parameters and syntax, see "wanboot.conf File Parameters and Syntax" on page 600.

▼ To Create a wanboot.conf File

1. **Assume the same user role as the web server user on the WAN boot server.**

2. **Create the wanboot.conf text file.**

   You can create a new text file that is named wanboot.conf, or use the sample file that is located in /etc/netboot/wanboot.conf.sample. If you use the sample file, rename the file wanboot.conf after you add parameters.

3. **Type the wanboot.conf parameters and values for your installation.**

   For detailed descriptions of wanboot.conf parameters and values, see "wanboot.conf File Parameters and Syntax" on page 600.

4. **Save the wanboot.conf file to the appropriate subdirectory of the /etc/netboot hierarchy.**

   For information about how to create the /etc/netboot hierarchy, see "Creating the /etc/netboot Hierarchy on the WAN Boot Server" on page 534.

5. **Validate the wanboot.conf file.**

   ```
   # bootconfchk /etc/netboot/path-to-wanboot.conf/wanboot.conf
   ```

path-to-wanboot.conf Specifies the path to the client's `wanboot.conf` file on the WAN boot server

- If the `wanboot.conf` file is structurally valid, the `bootconfchk` command returns an exit code of 0.
- If the `wanboot.conf` file is invalid, the `bootconfchk` command returns a nonzero exit code.

6. **Change the permissions on the `wanboot.conf` file to 600.**

```
# chmod 600 /etc/netboot/path-to-wanboot.conf/wanboot.conf
```

EXAMPLE 43–9 `wanboot.conf` File for WAN Boot Installation Over HTTPS

The following `wanboot.conf` file example includes configuration information for a WAN installation that uses secure HTTP. The `wanboot.conf` file also indicates that a 3DES encryption key is used in this installation.

```
boot_file=/wanboot/wanboot.s9_sparc
root_server=https://www.example.com:1234/cgi-bin/wanboot-cgi
root_file=/miniroot/miniroot.s9_sparc
signature_type=sha1
encryption_type=3des
server_authentication=yes
client_authentication=no
resolve_hosts=
boot_logger=https://www.example.com:1234/cgi-bin/bootlog-cgi
system_conf=system.conf
```

This `wanboot.conf` file specifies the following configuration.

`boot_file=/wanboot/wanboot.s9_sparc`
The second level boot program is named `wanboot.s9_sparc`. This program is located in the `/wanboot` directory in the WAN boot server's document root directory.

`root_server=https://www.example.com:1234/cgi-bin/wanboot-cgi`
The location of the `wanboot-cgi` program on the WAN boot server is `https://www.example.com:1234/cgi-bin/wanboot-cgi`. The `https` portion of the URL indicates that this WAN boot installation uses secure HTTP.

`root_file=/miniroot/miniroot.s9_sparc`
The WAN boot miniroot is named `miniroot.s9_sparc`. This miniroot is located in the `/miniroot` directory in the WAN boot server's document root directory.

`signature_type=sha1`
The `wanboot.s9_sparc` program and the WAN boot file system are signed with a HMAC SHA1 hashing key.

`encryption_type=3des`
The `wanboot.s9_sparc` program and the boot file system are encrypted with a 3DES key.

EXAMPLE 43–9 wanboot.conf File for WAN Boot Installation Over HTTPS *(Continued)*

server_authentication=yes
The server is authenticated during the installation.

client_authentication=no
The client is not authenticated during the installation.

resolve_hosts=
No additional host names are needed to perform the WAN installation. All required files and information are located in the document root directory on the WAN boot server.

boot_logger=https://www.example.com:1234/cgi-bin/bootlog-cgi
(Optional) Booting and installation log messages are recorded on the WAN boot server by using secure HTTP.

For instructions on how to set up a logging server for your WAN boot installation, see "(Optional) Configuring the WAN Boot Logging Server" on page 537.

system_conf=system.conf
The system configuration file that contains the locations of the sysidcfg and JumpStart files is located in a subdirectory of the /etc/netboot hierarchy. The system configuration file is named system.conf.

EXAMPLE 43–10 wanboot.conf File for Insecure WAN Boot Installation

The following wanboot.conf file example includes configuration information for a less secure WAN boot installation that uses HTTP. This wanboot.conf file also indicates that the installation does not use an encryption key or a hashing key.

```
boot_file=/wanboot/wanboot.s9_sparc
root_server=http://www.example.com/cgi-bin/wanboot-cgi
root_file=/miniroot/miniroot.s9_sparc
signature_type=
encryption_type=
server_authentication=no
client_authentication=no
resolve_hosts=
boot_logger=http://www.example.com/cgi-bin/bootlog-cgi
system_conf=system.conf
```

This wanboot.conf file specifies the following configuration.

boot_file=/wanboot/wanboot.s9_sparc
The second level boot program is named wanboot.s9_sparc. This program is located in the /wanboot directory in the WAN boot server's document root directory.

root_server=http://www.example.com/cgi-bin/wanboot-cgi
The location of the wanboot-cgi program on the WAN boot server is http://www.example.com/cgi-bin/wanboot-cgi. This installation does not use secure HTTP.

EXAMPLE 43–10 `wanboot.conf` File for Insecure WAN Boot Installation *(Continued)*

`root_file=/miniroot/miniroot.s9_sparc`
The WAN boot miniroot is named `miniroot.s9_sparc`. This miniroot is located in the `/miniroot` subdirectory in the WAN boot server's document root directory.

`signature_type=`
The `wanboot.s9_sparc` program and the WAN boot file system are not signed with a hashing key.

`encryption_type=`
The `wanboot.s9_sparc` program and the boot file system are not encrypted.

`server_authentication=no`
The server is not authenticated with keys or certificates during the installation.

`client_authentication=no`
The client is not authenticated with keys or certificates during the installation.

`resolve_hosts=`
No additional host names are needed to perform the installation. All required files and information are located in the document root directory on the WAN boot server.

`boot_logger=http://www.example.com/cgi-bin/bootlog-cgi`
(Optional) Booting and installation log messages are recorded on the WAN boot server.

For instructions on how to set up a logging server for your WAN boot installation, see "(Optional) Configuring the WAN Boot Logging Server" on page 537.

`system_conf=system.conf`
The system configuration file that contains the locations of the `sysidcfg` and JumpStart files is named `system.conf`. This file is located in the appropriate client subdirectory of the `/etc/netboot` hierarchy.

(Optional) Providing Configuration Information With a DHCP Server

If you use a DHCP server on your network, you can configure the DHCP server to supply the following information.

- Proxy server's IP address
- Location of the `wanboot-cgi` program

You can use the following DHCP vendor options in your WAN boot installation.

`SHTTPproxy` Specifies the IP address of the network's proxy server

`SbootURI` Specifies the URL of the `wanboot-cgi` program on the WAN boot server

For information about setting these vendor options on a Solaris DHCP server, see "Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85.

For detailed information about setting up a Solaris DHCP server, see "Configuring DHCP Service (Task)" in *System Administration Guide: IP Services*.

CHAPTER 44

SPARC: Installing With WAN Boot (Tasks)

This chapter describes how to perform a WAN boot installation on a SPARC based client. For information about how to prepare for a WAN boot installation, see Chapter 43.

This chapter describes the following tasks.

- "Preparing the Client for a WAN Boot Installation" on page 558
- "Installing the Client" on page 565

Task Map: Installing a Client With WAN Boot

The following table lists the tasks you need to perform to install a client over a WAN.

TABLE 44–1 Task Map: Performing a WAN Boot Installation

Task	Description	For Instructions
Prepare the network for a WAN boot installation.	Set up the servers and files that are required to perform a WAN boot installation.	Chapter 43
Verify that the client system supports WAN boot.	Check the client OBP for boot argument support of WAN boot.	"Checking the Client OBP for WAN Boot Support" on page 559

TABLE 44–1 Task Map: Performing a WAN Boot Installation *(Continued)*

Task	Description	For Instructions
Verify that the `net` device alias is set correctly in the client OBP.	Use the `devalias` command to verify that the `net` device alias is set to the primary network interface.	"Checking the `net` Device Alias in the Client OBP" on page 559
Provide keys to the client	Provide keys to the client by setting OBP variables or entering key values during the installation. This task is required for secure installation configurations. For insecure installations that check data integrity, complete this task to provide the HMAC SHA1 hashing key to the client.	"Installing Keys on the Client" on page 561
Install the client over a wide area network.	Choose the appropriate method to install your client.	"Performing a Noninteractive WAN Boot Installation" on page 567 "Performing an Interactive WAN Boot Installation" on page 568 "Installing With a DHCP Server" on page 572 "Installing With Local CD Media" on page 574

Preparing the Client for a WAN Boot Installation

Before you install the client system, prepare the client by performing the following tasks.

- "Checking the Client OBP for WAN Boot Support" on page 559
- "Checking the `net` Device Alias in the Client OBP" on page 559
- "Installing Keys on the Client" on page 561

Checking the Client OBP for WAN Boot Support

To perform a hands-off WAN boot installation, the client OpenBoot PROM (OBP) must support WAN boot. The following procedure describes how to determine if the client OBP supports WAN boot.

▼ To Check the Client OBP for WAN Boot Support

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

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

2. **Check the OBP configuration variables for WAN boot support.**

   ```
   # eeprom | grep network-boot-arguments
   ```

   - If the variable `network-boot-arguments` is displayed, or if the previous command returns the output `network-boot-arguments: data not available`, the OBP supports WAN boot installations. You do not need to update the OBP before you perform your WAN boot installation.
   - If the previous command does not return any output, the OBP does not support WAN boot installations. You must perform one of the following tasks.
     - Update the client OBP. See your system documentation for information about how to update the OBP.
     - Perform the WAN boot installation from the Solaris 9 Software CD in a local CD-ROM drive. For instructions about how to boot the client from a local CD-ROM drive, see "Installing With Local CD Media" on page 574.

EXAMPLE 44–1 Verifying OBP Support for WAN Boot on the Client

The following command shows how to check the client OBP for WAN boot support.

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

In this example, the output `network-boot-arguments: data not available` indicates that the client OBP supports WAN boot.

Checking the `net` Device Alias in the Client OBP

To boot the client from the WAN with the `boot net`, the `net` device alias must be set to the client's primary network device. On most systems, this alias is already set correctly. However, if the alias is not set to the network device you want to use, you must change the alias..

Follow these steps to check the `net` device alias on the client.

To Check the `net` Device Alias

1. **Become superuser on the client.**

2. **Bring the system to run level 0.**

   ```
    # init 0
   ```

   The `ok` prompt is displayed.

3. **At the `ok` prompt, check device aliases that are set in the OBP.**

   ```
   ok devalias
   ```

   The devalias command outputs information that is similar to the following example.

   ```
   screen                   /pci@1f,0/pci@1,1/SUNW,m64B@2
   net                      /pci@1f,0/pci@1,1/network@c,1
   net2                     /pci@1f,0/pci@1,1/network@5,1
   disk                     /pci@1f,0/pci@1/scsi@8/disk@0,0
   cdrom                    /pci@1f,0/pci@1,1/ide@d/cdrom@0,0:f
   keyboard                 /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
   mouse                    /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
   ```

   - If the `net` alias is set to the network device you wan to use during the installation, you do not need to reset the alias. Go to "Installing Keys on the Client" on page 561 to continue your installation.
   - If the `net` alias is not set to the network device you want to use, you must reset the alias. Continue.

4. **Set the `net` device alias.**

   Choose one of the following commands to set the `net` device alias.

   - To set the `net` device alias for this installation only, use the `devalias` command.

     ```
     ok devalias net device-path
     ```

     `net` *device-path*   Assigns the device *device-path* to the `net` alias

   - To permanently set the `net` device alias, use the `nvalias` command.

     ```
     ok nvalias net device-path
     ```

     `net` *device-path*   Assigns the device *device-path* to the `net` alias

EXAMPLE 44–2 Checking and Resetting the `net` Device Alias

The following commands show how to check and reset the `net` device alias.

Check the device aliases.

```
ok devalias
screen                   /pci@1f,0/pci@1,1/SUNW,m64B@2
net                      /pci@1f,0/pci@1,1/network@c,1
```

```
net2                    /pci@1f,0/pci@1,1/network@5,1
disk                    /pci@1f,0/pci@1/scsi@8/disk@0,0
cdrom                   /pci@1f,0/pci@1,1/ide@d/cdrom@0,0:f
keyboard                /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse                   /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
```

If you want to use the `/pci@1f,0/pci@1,1/network@5,1` network device, type the following command.

```
ok devalias net /pci@1f,0/pci@1,1/network@5,1
```

For more information about setting device aliases, see “The Device Tree” in *OpenBoot 3.x Command Reference Manual*.

Installing Keys on the Client

For a more secure WAN boot installation or an insecure installation with data integrity checking, you must install keys on the client. By using a hashing key and an encryption key, you can protect the data that is transmitted to the client. You can install these keys in the following ways.

- Set OBP variables – You can assign key values to OBP network boot argument variables before you boot the client. These keys can then be used for future WAN boot installations of the client.
- Enter the key values during the boot process – You can set key values at the `wanboot` program `boot>` prompt. If you use this method to install keys, the keys are only used for the current WAN boot installation.

You can also install keys in the OBP of a running client. If you want to install keys on a running client, the system must be running the Solaris 9 12/03 operating environment, or compatible version.

When you install keys on your client, ensure that the key values are not transmitted over an insecure connection. Follow your site’s security policies to ensure the privacy of the key values.

- For instructions about how to assign key values to OBP network boot argument variables, see “To Install Keys in the Client OBP” on page 562.
- For instructions about how to install keys during the boot process, see “To Perform an Interactive Installation” on page 569.
- For instructions about how to install keys in the OBP of a running client, see “To Install a Hashing Key and an Encryption Key on a Running Client” on page 564.

▼ To Install Keys in the Client OBP

If you want to assign key values to OBP network boot argument variables, follow these steps.

1. **Assume the same user role as the web server user on the WAN boot server.**

2. **Display the key value for each client key.**

   ```
   # wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
   ```

   net-ip The IP address of the client's subnet.

   client-ID The ID of the client you want to install. The client ID can be a user-defined ID or the DHCP client ID.

   key-type The key type you want to install on the client. Valid key types are `3des`, `aes`, or `sha1`.

   The hexadecimal value for the key is displayed.

3. **Repeat the previous step for each type of client key you want to install.**

4. **Bring the client system to run level 0.**

   ```
   # init 0
   ```

   The `ok` prompt is displayed.

5. **At the client `ok` prompt, set the value for the hashing key.**

   ```
   ok set-security-key wanboot-hmac-sha1 key-value
   ```

   `set-security-key` Installs the key on the client

   `wanboot-hmac-sha1` Instructs OBP to install a HMAC SHA1 hashing key

   key-value Specifies the hexadecimal string that is displayed in Step 2.

   The HMAC SHA1 hashing key is installed in the client OBP.

6. **At the client `ok` prompt, install the encryption key.**

   ```
   ok set-security-key wanboot-3des key-value
   ```

   `set-security-key` Installs the key on the client

   `wanboot-3des` Instructs OBP to install a 3DES encryption key. If you want to use an AES encryption key, set this value to `wanboot-aes`.

   key-value Specifies the hexadecimal string that represents the encryption key.

   The 3DES encryption key is installed in the client OBP.

After you install the keys, you are ready to install the client. See “Installing the Client” on page 565 for instructions about how to install the client system.

7. **(Optional) Verify that the keys are set in the client OBP.**

   ```
   ok list-security-keys
   Security Keys:
            wanboot-hmac-sha1
            wanboot-3des
   ```

8. **(Optional) If you need to delete a key, type the following command.**

   ```
   ok set-security-key key-type
   ```

   key-type Specifies the type of key you need to delete. Use the value `wanboot-hmac-sha1`, `wanboot-3des`, or `wanboot-aes`.

EXAMPLE 44–3 Installing Keys in the Client OBP

The following example shows how to install a hashing key and an encryption key in the client OBP.

Display the key values on the WAN boot server.

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous example uses the following information.

`net=192.168.198.0`
Specifies the IP address of the client’s subnet

`cid=010003BA152A42`
Specifies the client’s ID

`b482aaab82cb8d5631e16d51478c90079cc1d463`
Specifies the value of the client’s HMAC SHA1 hashing key

`9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04`
Specifies the value of the client’s 3DES encryption key

If you use an AES encryption key in your installation, change `wanboot-3des` to `wanboot-aes` to display the encryption key value.

Install the keys on the client system.

```
ok set-security-key wanboot-hmac-sha1 b482aaab82cb8d5631e16d51478c90079cc1d463
ok set-security-key wanboot-3des 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous commands perform the following tasks.

- Installs the HMAC SHA1 hashing key with a value of `b482aaab82cb8d5631e16d51478c90079cc1d463` on the client

- Installs the 3DES encryption key with a value of `9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04` on the client

  If you use an AES encryption key in your installation, change `wanboot-3des` to `wanboot-aes`.

▼ To Install a Hashing Key and an Encryption Key on a Running Client

If you want to install a hashing key and an encryption key in the OBP of a running client, follow these steps.

Note – This procedure makes the following assumptions.

- The client system is powered on.
- The client is accessible over a secure connection, such as a secure shell (`ssh`).

---

1. **Assume the same user role as the web server user on the WAN boot server.**

2. **Display the key value for the client keys.**

   ```
   # wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
   ```

   net-ip The IP address of the client's subnet.

   client-ID The ID of the client you want to install. The client ID can be a user-defined ID or the DHCP client ID.

   key-type The key type you want to install on the client. Valid key types are `3des`, `aes`, or `sha1`.

   The hexadecimal value for the key is displayed.

3. **Repeat the previous step for each type of client key you want to install.**

4. **Become superuser on the client machine.**

5. **Install the necessary keys on the running client machine.**

   ```
   # /usr/lib/inet/wanboot/ickey -o type=key-type
   > key-value
   ```

   key-type Specifies the key type you want to install on the client. Valid key types are `3des`, `aes`, or `sha1`.

   key-value Specifies the hexadecimal string that is displayed in Step 2.

6. **Repeat the previous step for each type of client key you want to install.**

After you install the keys, you are ready to install the client. See "Installing the Client" on page 565 for instructions about how to install the client system.

EXAMPLE 44–4 Installing Keys in the OBP of a Running Client System

The following example shows how to install keys in the OBP of a running client.

Display the key values on the WAN boot server.

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous example uses the following information.

`net=192.168.198.0`
Specifies the IP address of the client's subnet

`cid=010003BA152A42`
Specifies the client's ID

`b482aaab82cb8d5631e16d51478c90079cc1d463`
Specifies the value of the client's HMAC SHA1 hashing key

`9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04`
Specifies the value of the client's 3DES encryption key

If you use an AES encryption key in your installation, change `type=3des` to `type=aes` to display the encryption key value.

Install the keys in the OBP of the running client.

```
# /usr/lib/inet/wanboot/ickey -o type=sha1 b482aaab82cb8d5631e16d51478c90079cc1d463
# /usr/lib/inet/wanboot/ickey -o type=3des 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous commands perform the following tasks.

- Installs a HMAC SHA1 hashing key with a value of `b482aaab82cb8d5631e16d51478c90079cc1d463` on the client
- Installs a 3DES encryption key with a value of `9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04` on the client

Installing the Client

When you finish preparing your network for a WAN boot installation, you can choose from the following ways to install the system.

TABLE 44–2 Methods to Install the Client

Method	Description	Instructions
Noninteractive installation	Use this installation method if you want to install keys on the client and set the client configuration information before you boot the client.	■ To install keys on the client before the installation, see "Installing Keys on the Client" on page 561. ■ To perform a noninteractive installation, see "To Perform a Noninteractive Installation" on page 567.
Interactive installation	Use this installation method if you want to set the client configuration information during the boot process.	"To Perform an Interactive Installation" on page 569
Installing with a DHCP server	Use this installation method if you configured the network DHCP server to provide client configuration information during the installation.	■ To configure a DHCP server to support a WAN boot installation, see "(Optional) Providing Configuration Information With a DHCP Server" on page 555. ■ To use a DHCP server during your installation, see "To Install With a DHCP Server" on page 573.
Installing with local CD media	If your client OBP does not support WAN boot, boot the client from a local copy of the Solaris 9 9/04 Software CD.	■ To determine if the client OBP supports WAN boot, see "To Check the Client OBP for WAN Boot Support" on page 559. ■ To install the client with a local copy of the Solaris 9 9/04 Software CD, see "To Install With Local CD Media" on page 574.

Performing a Noninteractive WAN Boot Installation

Use this installation method if you prefer to install keys and set client configuration information before you install the client. You can then boot the client from the WAN and perform an unattended installation.

This procedure assumes that you have either installed keys in the client's OBP, or that you are performing an insecure installation. For information about installing keys on the client before your installation, see "Installing Keys on the Client" on page 561.

▼ To Perform a Noninteractive Installation

1. **If the client system is currently running, bring the system to run level 0.**

```
# init 0
```

The `ok` prompt is displayed.

2. **At the `ok` prompt on the client system, set the network boot argument variables in OBP.**

```
ok setenv network-boot-arguments
host-ip=client-IP,router-ip=router-ip,
subnet-mask=mask-value,hostname=client-name,
http-proxy=proxy-ip:port,file=wanbootCGI-URL
```

Note – The line breaks in this command sample are included for formatting purposes only. Do not enter a carriage return until you finish typing the command.

`setenv network-boot-arguments`	Instructs the OBP to set the following boot arguments
`host-ip=`*client-IP*	Specifies the IP address of the client
`router-ip=`*router-ip*	Specifies the IP address of the network router
`subnet-mask=`*mask-value*	Specifies the subnet mask value
`hostname=`*client-name*	Specifies the host name of the client
(Optional) `http-proxy=`*proxy-ip:port*	Specifies the IP address and port of the network's proxy server
`file=`*wanbootCGI-URL*	Specifies the URL of the `wanboot-cgi` program on the web server

3. **Boot the client.**

```
ok boot net - install
```

`net - install` Instructs the client to use the network boot argument variables to boot from the WAN

The client installs over the WAN. If the WAN boot programs do not find all the necessary installation information, the wanboot program prompts to provide the missing information. Type the additional information at the prompt.

EXAMPLE 44–5 Noninteractive WAN Boot Installation

In the following example, the network boot argument variables for the client system seahag are set before the machine is booted. This example assumes that a hashing key and encryption key are already installed on the client. For information about installing keys before you boot from the WAN, see "Installing Keys on the Client" on page 561.

```
ok setenv network-boot-arguments host-ip=192.168.198.136,
router-ip=192.168.198.129,subnet-mask=255.255.255.192,hostname=seahag,
file=http://192.168.198.135/cgi-bin/wanboot-cgi

ok boot net - install
Resetting ...




Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.




Rebooting with command: boot net - install
Boot device: /pci@1f,0/network@c,1  File and args: - install
```

The following variables are set.

- The client IP address is set to 192.168.198.136.
- The client's router IP address is set to 192.168.198.129.
- The client's subnet mask is set to 255.255.255.192.
- The client's host name is set to seahag.
- The wanboot-cgi program is located at http://192.168.198.135/cgi-bin/wanboot-cgi.

Performing an Interactive WAN Boot Installation

Use this installation method if you want to install keys and set client configuration information at the command line during the installation.

▼ To Perform an Interactive Installation

This procedure assumes that you are using HTTPS in your WAN installation. If you are performing an insecure installation that does not use keys, do not display or install the client keys.

1. **Assume the same user role as the web server user on the WAN boot server.**
2. **Display the key value for each client key.**

   ```
   # wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
   ```

   net-ip The IP address of the subnet for the client you want to install.

   client-ID The ID of the client you want to install. The client ID can be a user-defined ID or the DHCP client ID.

   key-type The key type you want to install on the client. Valid key types are `3des`, `aes`, or `sha1`.

   The hexadecimal value for the key is displayed.
3. **Repeat the previous step for each type of client key you are installing.**
4. **If the client system is currently running, bring the client to run level 0.**
5. **At the `ok` prompt on the client system, set the network boot argument variables in OBP.**

   ```
   ok setenv network-boot-arguments
   host-ip=client-IP,router-ip=router-ip,
   subnet-mask=mask-value,hostname=client-name,
   http-proxy=proxy-ip:port,bootserver=wanbootCGI-URL
   ```

   Note – The line breaks in this command sample are included for formatting purposes only. Do not enter a carriage return until you finish typing the command.

`setenv network-boot-arguments`	Instructs the OBP to set the following boot arguments
`host-ip=`*client-IP*	Specifies the IP address of the client
`router-ip=`*router-ip*	Specifies the IP address of the network router
`subnet-mask=`*mask-value*	Specifies the subnet mask value
`hostname=`*client-name*	Specifies the host name of the client
(Optional) `http-proxy=`*proxy-ip:port*	Specifies the IP address and port of the network's proxy server

bootserver=*wanbootCGI-URL* Specifies the URL of the wanboot-cgi program on the web server

---

Note – The URL value for thebootserver variable must not be an HTTPS URL. The URL must start with http://.

---

6. **At the client ok prompt, boot the system.**

```
ok boot net -o prompt - install
```

net -o prompt - install Instructs the client to boot and install from the network. The wanboot program prompts the user to enter client configuration information at the boot> prompt.

The boot> prompt is displayed.

7. **Install the encryption key.**

```
boot> 3des=key-value
```

3des=*key-value* Specifies the hexadecimal string of the 3DES key that is displayed in Step 2.

If you use an AES encryption key, use the following format for this command.

```
boot> aes=key-value
```

8. **Install the hashing key.**

```
boot> sha1=key-value
```

sha1=*key-value* Specifies the hashing key value that is displayed in Step 2.

9. **Type the following command to continue the boot process.**

```
boot> go
```

The client installs over the WAN.

10. **If prompted, type client configuration information on the command line.**

If the WAN boot programs do not find all the necessary installation information, the wanboot program prompts to provide the missing information. Type the additional information at the prompt.

EXAMPLE 44–6 Interactive WAN Boot Installation

In the following example, the wanboot program prompts you to set the key values for the client system during the installation.

Display the key values on the WAN boot server.

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous example uses the following information.

net=192.168.198.0
  Specifies the IP address of the client's subnet

cid=010003BA152A42
  Specifies the client's ID

b482aaab82cb8d5631e16d51478c90079cc1d463
  Specifies the value of the client's HMAC SHA1 hashing key

9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
  Specifies the value of the client's 3DES encryption key

  If you use an AES encryption key in your installation, change type=3des to type=aes to display the encryption key value.

Set the network boot argument variables in the OBP on the client.

```
ok setenv network-boot-arguments host-ip=192.168.198.136,
router-ip=192.168.198.129,subnet-mask=255.255.255.192,hostname=seahag,
bootserver=http://192.168.198.135/cgi-bin/wanboot-cgi
```

The following variables are set.

- The client IP address is set to 192.168.198.136.
- The client's router IP address is set to 192.168.198.129.
- The client's subnet mask is set to 255.255.255.192.
- The client's host name is set to seahag.
- The wanboot-cgi program is located at http://192.168.198.135/cgi-bin/wanboot-cgi.

Boot and install the client.

```
ok boot net -o prompt - install
Resetting ...


Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
```

EXAMPLE 44–6 Interactive WAN Boot Installation *(Continued)*

```
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.



Rebooting with command: boot net -o prompt
Boot device: /pci@1f,0/network@c,1  File and args: -o prompt

boot> 3des=9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04

boot> sha1=b482aaab82cb8d5631e16d51478c90079cc1d463

boot> go
```

The previous commands perform the following tasks.

- Installs the 3DES encryption key with the value `9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04` on the client
- Installs the HMAC SHA1 hashing key with the value `b482aaab82cb8d5631e16d51478c90079cc1d463` on the client
- Starts the installation

Installing With a DHCP Server

If you configured a DHCP server to support WAN boot options, you can use the DHCP server to provide client configuration information during the installation. For more information about configuring a DHCP server to support a WAN boot installation, see "(Optional) Providing Configuration Information With a DHCP Server" on page 555.

This procedure makes the following assumptions.

- The client system is running.
- You have either installed keys on the client, or you are performing an insecure installation.

  For information about installing keys on the client before your installation, see "Installing Keys on the Client" on page 561.
- You have configured your DHCP server to support the `SbootURI` and `SHTTPproxy` WAN boot options.

  These options enable the DHCP server to provide the configuration information that is required by WAN boot.

  For information about how to set installation options on your DHCP server, see "Preconfiguring System Configuration Information With the DHCP Service (Tasks)" on page 85.

▼ To Install With a DHCP Server

1. **If the client system is currently running, bring the system to run level 0.**

   ```
   # init 0
   ```

   The ok prompt is displayed.

2. **At the ok prompt on the client system, set the network boot argument variables in OBP.**

   ```
   ok setenv network-boot-arguments dhcp,hostname=client-name
   ```

   `setenv network-boot-arguments` Instructs the OBP to set the following boot arguments

   `dhcp` Instructs the OBP to use the DHCP server to configure the client

   `hostname=`*client-name* Specifies the host name you want to assign to the client

3. **Boot the client from the network.**

   ```
   ok boot net - install
   ```

   `net - install` Instructs the client to use the network boot argument variables to boot from the WAN

   The client installs over the WAN. If the WAN boot programs do not find all the necessary installation information, the wanboot program prompts to provide the missing information. Type the additional information at the prompt.

EXAMPLE 44–7 WAN Boot Installation With a DHCP Server

In the following example, the DHCP server on the network provides client configuration information. This sample requests the host name myhost for the client.

```
ok setenv network-boot-arguments dhcp,
hostname=myhost

ok boot net - install
Resetting ...



Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.



Rebooting with command: boot net - install
Boot device: /pci@1f,0/network@c,1  File and args: - install
```

Installing With Local CD Media

If your client's OBP does not support WAN boot, you can install with a Solaris Software 1 of 2 CD inserted in the client's CD-ROM drive. When you use a local CD, the client retrieves the `wanboot` program from the local media, rather than from the WAN boot server.

Follow these steps to perform a WAN boot installation from a local CD.

▼ To Install With Local CD Media

This procedure assumes that you are using HTTPS in your WAN installation. If you are performing an insecure installation, do not display or install the client keys.

1. **Assume the same user role as the web server user on the WAN boot server.**

2. **Display the key value for each client key.**

   ```
   # wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
   ```

   net-ip The network IP address for the client you are installing.

   client-ID The ID of the client you are installing. The client ID can be a user-defined ID or the DHCP client ID.

   key-type The key type you are installing on the client. Valid key types are `3des`, `aes`, or `sha1`.

   The hexadecimal value for the key is displayed.

3. **Repeat the previous step for each type of client key you are installing.**

4. **On the client system, insert the Solaris Software 1 of 2 CD in the CD-ROM drive.**

5. **Power on the client system.**

6. **Boot the client from the CD.**

   ```
   ok boot cdrom -o prompt -F wanboot - install
   ```

   `cdrom` Instructs the OBP to boot from the local CD-ROM

   `-o prompt` Instructs the `wanboot` program to prompt the user to enter client configuration information

   `-F wanboot` Instructs the OBP to load the `wanboot` program from the CD-ROM

   `- install` Instructs the client to perform a WAN boot installation

   The client's OBP loads the `wanboot` program from the Solaris Software 1 of 2 CD. The `wanboot` program boots the system, and the `boot>` prompt is displayed.

7. **Type the encryption key value.**

   ```
   boot> 3des=key-value
   ```

   `3des=`*key-value* Specifies the hexadecimal string of the 3DES key that is displayed in step Step 2.

   If you use an AES encryption key, use the following format for this command.

   ```
   boot> aes=key-value
   ```

8. **Type the hashing key value.**

   ```
   boot> sha1=key-value
   ```

   `sha1=`*key-value* Specifies the hexadecimal string that represents the hashing key value that is displayed in step Step 2.

9. **Set the network interface variables.**

   ```
   boot> variable=value[,variable=value*]
   ```

   Type the following variable and value pairs at the `boot>` prompt.

   `host-ip=`*client-IP* Specifies the IP address of the client.

   `router-ip=`*router-ip* Specifies the IP address of the network router.

   `subnet-mask=`*mask-value* Specifies the subnet mask value.

   `hostname=`*client-name* Specifies the host name of the client.

   (Optional) `http-proxy=`*proxy-ip:port* Specifies the IP address and port number of the network's proxy server.

   `bootserver=`*wanbootCGI-URL* Specifies the URL of the `wanboot-cgi` program on the web server.

   Note – The URL value for the`bootserver` variable must not be an HTTPS URL. The URL must start with `http://`.

   You can enter these variables in the following ways.

   - Type one variable and value pair at the `boot>` prompt, then press the Return key.

     ```
     boot> host-ip=client-IP
     boot> subnet-mask=mask-value
     ```

- Type all the variable and value pairs on one `boot>` prompt line, then press the Return key. Type commas to separate each variable and value pair.

  ```
  boot> host-ip=client-IP,subnet-mask=mask-value,
  router-ip=router-ip,hostname=client-name,
  http-proxy=proxy-ip:port,bootserver=wanbootCGI-URL
  ```

10. Type the following command to continue the boot process.

```
boot> go
```

The client installs over the WAN. If the WAN boot programs do not find all the necessary installation information, the `wanboot` program prompts to provide the missing information. Type the additional information at the prompt.

EXAMPLE 44–8 Installing With Local CD Media

In the following example, the `wanboot` program on a local CD prompts you to set the network interface variables for the client during the installation.

Display the key values on the WAN boot server.

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous example uses the following information.

`net=192.168.198.0`
Specifies the IP address of the client's subnet

`cid=010003BA152A42`
Specifies the client's ID

`b482aaab82cb8d5631e16d51478c90079cc1d463`
Specifies the value of the client's HMAC SHA1 hashing key

`9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04`
Specifies the value of the client's 3DES encryption key

If you use an AES encryption key in your installation, change `type=3des` to `type=aes` to display the encryption key value.

Boot and install the client.

```
ok boot cdrom -o prompt -F wanboot - install
Resetting ...


Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.
```

EXAMPLE 44–8 Installing With Local CD Media *(Continued)*

```
Rebooting with command: boot cdrom -F wanboot - install
Boot device: /pci@1f,0/network@c,1  File and args: -o prompt

boot> 3des=9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04

boot> sha1=b482aaab82cb8d5631e16d51478c90079cc1d463

boot> host-ip=192.168.198.124

boot> subnet-mask=255.255.255.128

boot> router-ip=192.168.198.1

boot> hostname=myhost

boot> client-id=010003BA152A42

boot> bootserver=http://192.168.198.135/cgi-bin/wanboot-cgi

boot> go
```

The previous commands perform the following tasks.

- Enters the 3DES encryption key with the value `9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04` on the client
- Enters the HMAC SHA1 hashing key with the value `b482aaab82cb8d5631e16d51478c90079cc1d463` on the client
- Sets the client IP address to 192.168.198.124
- Sets the client's subnet mask to 255.255.255.128
- Sets the client's router IP address to 192.168.198.1
- Sets the client's host name to `myhost`
- Sets the client ID to 010003BA152A42
- Sets the location of the `wanboot-cgi` program to `http://192.168.198.135/cgi-bin/wanboot-cgi/`

CHAPTER 45

SPARC: Installing With WAN Boot (Examples)

This chapter provides an example of setting up and installing client systems over a wide area network (WAN). The examples in this chapter describe how to perform a secure WAN boot installation over an HTTPS connection.

Sample Site Setup

Figure 45–1 shows the site setup for this example.

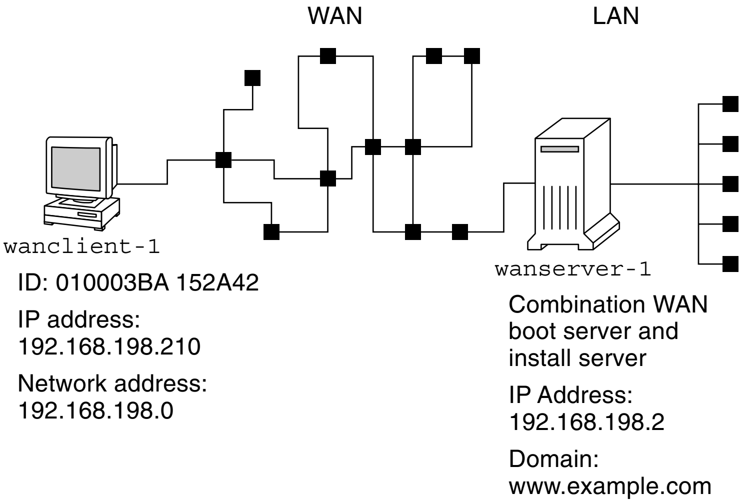


FIGURE 45–1 Sample Site for WAN Boot Installation

This sample site has the following characteristics.

- The server `wanserver-1` is to be configured as a WAN boot server and an install server.
- The IP address of `wanserver-1` is 192.168.198.2.
- The domain name of `wanserver-1` is `www.example.com`.
- `wanserver-1` is running the Solaris 9 9/04 operating environment.
- `wanserver-1` is running the Apache web server. The Apache software on `wanserver-1` is configured to support HTTPS.
- The client to be installed is named `wanclient-1`.
- `wanclient-1` is an UltraSPARCII system.
- The client ID for `wanclient-1` is 010003BA152A42.
- The IP address of `wanclient-1` is 192.168.198.210.
- The IP address of the client's subnet is 192.168.198.0.
- The client system `wanclient-1` has Internet access, but is not directly connected to the network that includes `wanserver-1`.

- wanclient-1 is a new system that is to be installed with the Solaris 9 9/04 operating environment.

Create the Document Root Directory

To store the installation files and data, set up the following directories in the document root directory (/opt/apache/htdocs) on wanserver-1.

- Solaris Flash directory

  ```
  wanserver-1# mkdir -p /opt/apache/htdocs/flash/
  ```

- WAN boot miniroot directory

  ```
  wanserver-1# mkdir -p /opt/apache/htdocs/miniroot/
  ```

- wanboot program directory

  ```
  wanserver-1# mkdir -p /opt/apache/htdocs/wanboot/
  ```

Create the WAN Boot Miniroot

Use the setup_install_server(1M) with the -w option to copy the WAN boot miniroot and the Solaris software image to the /export/install/Solaris_9 directory of wanserver-1.

Insert the Solaris Software media in the media drive that is attached to wanserver-1. Type the following commands.

```
wanserver-1# mkdir -p /export/install/sol_9_sparc
wanserver-1# cd /cdrom/cdrom0/s0/Solaris_9/Tools
wanserver-1# ./setup_install_server -w /export/install/sol_9_sparc/miniroot \
  /export/install/sol_9_sparc
```

Move the WAN boot miniroot to the document root directory (/opt/apache/htdocs/) of the WAN boot server.

```
wanserver-1# mv /export/install/sol_9_sparc/miniroot/miniroot \
  /opt/apache/htdocs/miniroot/miniroot.s9_sparc
```

Install the wanboot Program on the WAN Boot Server

To install the wanboot program on the WAN boot server, copy the program from the Solaris 9 9/04 Software media to the WAN boot server's document root directory.

Insert the Solaris DVD or the Solaris Software 1 of 2 CD in the media drive that is attached to wanserver-1 and type the following commands.

```
wanserver-1# cd /cdrom/cdrom0/s0/Solaris_9/Tools/Boot/platform/sun4u/
wanserver-1# cp wanboot /opt/apache/htdocs/wanboot/wanboot.s9_sparc
```

Create the /etc/netboot Hierarchy

Create the wanclient-1 subdirectories of the /etc/netboot directory on the WAN boot server. The WAN boot installation programs retrieve configuration and security information from this directory during the installation.

wanclient-1 is located on the subnet 192.168.198.0, and has a client ID of 010003BA152A42. To create the appropriate subdirectory of /etc/netboot for wanclient-1, perform the following tasks.

- Create the /etc/netboot directory.
- Change the permissions of the /etc/netboot directory to 700.
- Change the ownership of the /etc/netboot directory to the owner of the web server process.
- Assume the same user role as the web server user.
- Create a subdirectory of /etc/netboot that is named after the subnet (192.168.198.0).
- Create a subdirectory of the subnet directory that is named after the client ID.
- Change the permissions of the /etc/netboot subdirectories to 700.

```
wanserver-1# cd /
wanserver-1# mkdir /etc/netboot/
wanserver-1# chmod 700 /etc/netboot
wanserver-1# chown nobody:admin /etc/netboot
wanserver-1# exit
wanserver-1# su nobody
Password:
nobody# mkdir -p /etc/netboot/192.168.198.0/010003BA152A42
nobody# chmod 700 /etc/netboot/192.168.198.0
```

```
nobody# chmod 700 /etc/netboot/192.168.198.0/010003BA152A42
```

Copy the wanboot-cgi Program to the WAN Boot Server

On systems that are running the Solaris 9 9/04 operating environment, the wanboot-cgi program is located in the /usr/lib/inet/wanboot/ directory. To enable the WAN boot server to transmit the installation data, copy the wanboot-cgi program to the cgi-bin directory in the web server software directory.

```
wanserver-1# cp /usr/lib/inet/wanboot/wanboot-cgi \
  /opt/apache/cgi-bin/wanboot-cgi
wanserver-1# chmod 755 /opt/apache/cgi-bin/wanboot-cgi
```

(Optional) Configure the WAN Boot Server as a Logging Server

To view the boot and installation messages on the WAN boot server, copy the bootlog-cgi script to the cgi-bin directory on wanserver-1.

```
wanserver-1# cp /usr/lib/inet/wanboot/bootlog-cgi /opt/apache/cgi-bin/
wanserver-1# chmod 755 /opt/apache/cgi-bin/bootlog-cgi
```

Configure the WAN Boot Server to Use HTTPS

To use HTTPS in your WAN boot installation, you must enable SSL support in the web server software. You must also install a digital certificate on the WAN boot server. This example assumes that the Apache web server on wanserver-1 is configured to use SSL. This example also assumes that a digital certificate and a certificate authority that establish the identity of wanserver-1 are already installed on wanserver-1.

For examples about how to configure your web server software to use SSL, see you web server documentation.

Provide the Trusted Certificate to the Client

By requiring the server to authenticate itself to the client, you protect the data that is transmitted from the server to the client over HTTPS. To enable server authentication, you provide a trusted certificate to the client. The trusted certificate enables the client to verify the identity of the server during the installation.

To provide the trusted certificate to the client, assume the same user role as the web server user. Then, split the certificate to extract a trusted certificate. Then, insert the trusted certificate in the client's truststore file in the /etc/netboot hierarchy.

In this example, you assume the web server user role of nobody. Then, you split the server PKCS#12 certificate that is named cert.p12, and insert the trusted certificate in /etc/netboot directory for wanclient-1.

```
wanserver-1# su nobody
Password:
wanserver-1# wanbootutil p12split -i cert.p12 -t \
  /etc/netboot/192.168.198.0/010003BA152A42/truststore
```

(Optional) Use Private Key and Certificate for Client Authentication

To further protect your data during the installation, you might want to require wanclient-1 to authenticate itself to wanserver-1. To enable client authentication in your WAN boot installation, insert a client certificate and private key in the client subdirectory of the /etc/netboot hierarchy.

To provide a private key and certificate to the client, perform the following tasks.

- Assume the same user role as the web server user
- Split the PKCS#12 file into a private key and a client certificate
- Insert the certificate in the client's certstore file
- Insert the private key in the client's keystore file

In this example, you assume the web server user role of nobody. Then, you split the server PKCS#12 certificate that is named cert.p12. You insert certificate in the /etc/netboot hierarchy for wanclient-1. You then insert the private key that you named wanclient.key in the client's keystore file.

```
wanserver-1# su nobody
Password:
wanserver-1# wanbootutil p12split -i cert.p12 -c \
  /etc/netboot/192.168.198.0/010003BA152A42/certstore -k wanclient.key
wanserver-1# wanbootutil keymgmt -i -k wanclient.key \
  -s  /etc/netboot/192.168.198.0/010003BA152A42/keystore \
  -o type=rsa
```

Create the Keys for the Server and the Client

To protect the data transmitted between the server and client, you create a hashing key and an encryption key. The server uses the hashing key to protect the integrity of the wanboot program. The server uses the encryption key to encrypt the configuration and installation data. The client uses the hashing key to check the integrity of the downloaded wanboot program. The client uses the encryption key to decrypt the data during the installation.

First, you assume the same user role as the web server user. In this example, the web server user role is nobody.

```
wanserver-1# su nobody
Password:
```

Then, you use the wanbootutil keygen command to create a master HMAC SHA1 key for wanserver-1.

```
wanserver-1# wanbootutil keygen -m
```

Then, create a hashing key and an encryption key for wanclient-1.

```
wanserver-1# wanbootutil keygen -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
wanserver-1# wanbootutil keygen -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
```

The previous command creates a HMAC SHA1 hashing key and a 3DES encryption key for wanclient-1. 192.168.198.0 specifies the subnet of wanclient-1, and 010003BA152A42 specifies the client ID of wanclient-1.

Create the Solaris Flash Archive

In this example, you create your Solaris Flash archive by cloning the wanserver-1 master system. The archive is named sol-9-sparc, and is copied exactly from the master system. The archive is an exact duplicate of the master system. The archive is stored in sol-9-sparc.flar. You save the archive in the flash/archives subdirectory of the document root directory on the WAN boot server.

```
wanserver-1# flar create -n sol-9-sparc /opt/apache/htdocs/flash/archives/sol-9-sparc.flar
```

Create the sysidcfg File

To preconfigure the wanclient-1 system, specify keywords and values in the sysidcfg file. Save this file in the flash subdirectory of the document root directory of wanserver-1.

EXAMPLE 45–1 sysidcfg File for client-1 System

The following is an example of a sysidcfg file for wanclient-1. The host name, IP address, and netmask of these systems have been preconfigured by editing the name service. This file is located in the /opt/apache/htdocs/flash/sol_9_sparc directory.

```
network_interface=primary {hostname=wanclient-1
                           default_route=192.168.198.1
                           ip_address=192.168.198.210
                           netmask=255.255.255.0
                           protocol_ipv6=no}
timezone=US/Central
system_locale=C
terminal=xterm
timeserver=localhost
name_service=NIS {name_server=matter(192.168.254.254)
                  domain_name=leti.example.com
                  }
security_policy=none
```

Create the Client's Profile

For the `wanclient-1` system, create a profile that is named `wanclient_1_prof`. The `wanclient_1_prof` file contains the following entries, which define the Solaris 9 software to be installed on the `wanclient-1` system.

```
# profile keywords          profile values
# ----------------          -------------------
install_type                flash_install
archive_location            https://192.168.198.2/htdocs/flash/sol_9_sparc/archive1.flar
partitioning                explicit
filesys                     c0t1d0s0 4000 /
filesys                     c0t1d0s1 512 swap
filesys                     c0t1d0s7 free /export/home
```

The following list describes some of the keywords and values from this example.

`install_type` The profile installs a Solaris Flash archive on the clone system. All files are overwritten as in an initial installation.

`archive_location` The compressed Solaris Flash archive is retrieved from `wanserver-1`.

`partitioning` The file system slices are determined by the `filesys` keywords, value `explicit`. The size of root (/) is based on the size of the Solaris Flash archive. The size of `swap` is set to the necessary size and is installed on `c0t1d0s1`. `/export/home` is based on the remaining disk space. `/export/home` is installed on `c0t1d0s7`.

Create and Validate the `rules` File

The custom JumpStart programs use the the `rules` file to select the correct installation profile for the `wanclient-1` system. Create a text file that is named `rules`. Then, add keywords and values to this file.

The IP address of the `wanclient-1` system is 192.168.198.210. Use the `network` rule keyword to specify the profile that the custom JumpStart programs should use to install `client-1`.

```
network 192.168.198.210 - wanclient_1_prof -
```

This `rules` file instructs the custom JumpStart programs to use the `wanclient_1_prof` to install the Solaris 9 operating environment on `wanclient-1`.

Name this rule file wanclient_rule.

After you create the profile and the rules file, you run the check script to verify that the files are valid.

```
wanserver-1# ./check -r wanclient_rule
```

If the check script does not find any errors, the script creates the rules.ok file.

Save the rules.ok file in the /opt/apache/htdocs/flash/ directory.

Create the System Configuration File

Create a system configuration file that lists the locations of the sysidcfg file and the custom JumpStart files on the install server. Save this file in a directory that is accessible to the WAN boot server.

In the following example, the wanboot-cgi program looks for the sysidcfg and custom JumpStart files in the document root directory of the WAN boot server. The domain name of the WAN boot server is https://www.example.com. The WAN boot server is configured to use secure HTTP, so the data and files are protected during the installation.

In this example, the system configuration file is named sys.conf, and the file is saved in the /etc/netboot hierarchy on the WAN boot server. The sysidcfg and custom JumpStart files are located in the Solaris Flash subdirectory flash of the document root directory.

```
SsysidCF=https://www.example.com/htdocs/flash/
SjumpsCF=https://www.example.com/htdocs/flash/
```

Create the wanboot.conf File

WAN boot uses the configuration information that is included in the wanboot.conf file to install the client machine. Create the wanboot.conf file in a text editor. Save the file to the appropriate client subdirectory in the /etc/netboot hierarchy on the WAN boot server.

The following wanboot.conf file for wanclient-1 includes configuration information for a WAN installation that uses secure HTTP. This file also instructs WAN boot to use a HMAC SHA1 hashing key and a 3DES encryption key to protect data.

```
boot_file=/wanboot/wanboot.s9_sparc
root_server=https://www.example.com/cgi-bin/wanboot-cgi
root_file=/miniroot/miniroot.s9_sparc
signature_type=sha1
encryption_type=3des
server_authentication=yes
client_authentication=no
resolve_hosts=
boot_logger=
system_conf=sys.conf
```

This wanboot.conf file specifies the following configuration.

boot_file=/wanboot/wanboot.s9_sparc
The wanboot program is named wanboot.s9_sparc. This program is located in the wanboot directory in the document root directory on wanserver-1.

root_server=https://www.example.com/cgi-bin/wanboot-cgi
The location of the wanboot-cgi program on wanserver-1 is https://www.example.com/cgi-bin/wanboot-cgi. The https portion of the URL indicates that this WAN boot installation uses secure HTTP.

root_file=/miniroot/miniroot.s9_sparc
The WAN boot miniroot is named miniroot.s9_sparc. The miniroot is located in the miniroot directory in the document root directory on wanserver-1.

signature_type=sha1
The wanboot program and the WAN boot file system are signed by using a HMAC SHA1 hashing key.

encryption_type=3des
The wanboot program and the WAN boot file system are encrypted with a 3DES key.

server_authentication=yes
The server is authenticated during the installation.

client_authentication=no
The client is not authenticated during the installation.

Note – If you performed the tasks in "(Optional) Use Private Key and Certificate for Client Authentication" on page 584, set this parameter as client_authentication=yes `resolve_hosts=`

No additional host names are needed to perform the WAN installation. All the host names that are required by the `wanboot-cgi` program are specified in the `wanboot.conf` file and the client certificate.

`boot_logger=`

Booting and installation log messages are displayed on the system console. If you configured the logging server in "(Optional) Configure the WAN Boot Server as a Logging Server" on page 583, and you want WAN boot messages to appear on the WAN boot server as well, set this parameter to `boot_logger=https://www.example.com/cgi-bin/bootlog-cgi`.

`system_conf=sys.conf`

The system configuration file that specifies the locations of the `sysid.cfg` and JumpStart files is located in the `sys.conf` file in the `/etc/netboot` hierarchy on `wanserver-1`.

In this example, you save the `wanboot.conf` file in the `/etc/netboot/192.168.198.0/010003BA152A42` directory on `wanserver-1`.

Check the Client OBP for WAN Boot Support

Determine that the client OBP supports WAN boot by typing the following command on the client system.

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

In the previous example, the `network-boot-arguments: data not available` output indicates that the client OBP supports WAN boot.

Check the `net` Device Alias in OBP

To boot the client from the WAN with the `boot net`, the `net` device alias must be set to the client's primary network device. At the client `ok` prompt, type the `devalias` command to verify that the `net` alias is set to the primary network device `/pci@1f,0/pci@1,1/network@c,1`.

```
ok devalias
screen                    /pci@1f,0/pci@1,1/SUNW,m64B@2
net                       /pci@1f,0/pci@1,1/network@c,1
```

```
net2                    /pci@1f,0/pci@1,1/network@5,1
disk                    /pci@1f,0/pci@1/scsi@8/disk@0,0
cdrom                   /pci@1f,0/pci@1,1/ide@d/cdrom@0,0:f
keyboard                /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse                   /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
```

In the previous output example, the primary network device `/pci@1f,0/pci@1,1/network@c,1` is assigned to the `net` alias. You do not need to reset the alias.

Install Keys on the Client

In "Create the Keys for the Server and the Client" on page 585, you created the hashing key and encryption key to protect your data during the installation. To enable the client to decrypt the data transmitted from `wanserver-1` during the installation, install these keys on `wanclient-1`.

On `wanserver-1`, display the key values.

```
wanserver-1# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
wanserver-1# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous example uses the following information.

`net=192.168.198.0`
Specifies the IP address of the client's subnet

`cid=010003BA152A42`
Specifies the client's ID

`b482aaab82cb8d5631e16d51478c90079cc1d463`
Specifies the value of the client's HMAC SHA1 hashing key

`9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04`
Specifies the value of the client's 3DES encryption key

If you use an AES encryption key in your installation, change `type=3des` to `type=aes` to display the encryption key value.

At the `ok` prompt on `wanclient-1`, install the keys.

```
ok set-security-key wanboot-hmac-sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
ok set-security-key wanboot-3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

The previous commands perform the following tasks.

- Installs the HMAC SHA1 hashing key with a value of b482aaab82cb8d5631e16d51478c90079cc1d463 on wanclient-1
- Installs the 3DES encryption key with a value of 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04 on wanclient-1

Install the Client

You can perform an unattended installation by setting network boot argument variables for wanclient-1 at the ok prompt, and then booting the client.

```
ok setenv network-boot-arguments host-ip=192.168.198.210,
router-ip=192.168.198.1,subnet-mask=255.255.255.0,hostname=wanclient-1,
file=http://192.168.198.2/cgi-bin/wanboot-cgi

ok boot net - install
Resetting ...




Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.




Rebooting with command: boot net - install
Boot device: /pci@1f,0/network@c,1  File and args: - install




<time unavailable> wanboot progress: wanbootfs: Read 68 of 68 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Fri Jun 20 09:16:06 wanboot progress: miniroot: Read 166067 of 166067 kB (100%)
Fri Jun 20Tue Apr 15 09:16:06 wanboot info: miniroot: Download complete
SunOS Release 5.9 Version WANboot10:04/11/03 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Configuring devices.
```

The following variables are set.

- The client IP address is set to 192.168.198.210.
- The client's router IP address is set to 192.168.198.1
- The client's subnet mask is set to 255.255.255.0
- The client's host name is set to wanclient-1
- The wanboot-cgi program is located at http://192.168.198.2/cgi-bin/wanboot-cgi

The client installs over the WAN. If the `wanboot` program does not find all the necessary installation information, you might be prompted to provide the missing information at the command line.

CHAPTER 46

WAN Boot (Reference)

This chapter briefly describes the commands and files you use to perform a WAN installation.

- "WAN Boot Installation Commands" on page 595
- "OBP Commands" on page 598
- "System Configuration File Settings and Syntax" on page 599
- "wanboot.conf File Parameters and Syntax" on page 600

WAN Boot Installation Commands

The following tables describe the commands you use to perform a WAN boot installation.

- Table 46–1
- Table 46–2

TABLE 46–1 Preparing the WAN Boot Installation and Configuration Files

Task and Description	Command
Copy the Solaris installation image to *install-dir-path*, and copy the WAN boot miniroot to *wan-dir-path* on the install server's local disk.	`setup_install_server -w` *wan-dir-path install-dir-path*

TABLE 46–1 Preparing the WAN Boot Installation and Configuration Files *(Continued)*

Task and Description	Command
Create a Solaris Flash archive that is named *name*.`flar`. - *name* is the name of the archive - *optional-parameters* are optional parameters you can use to customize the archive - *document-root* is the path to the document root directory on the install server - *filename* is the name of the archive	`flar create – n` *name* [*optional-parameters*] *document-root*`/flash/`*filename*
Check the validity of the custom JumpStart `rules` file that is named *rules*.	`./check -r` *rules*
Check the validity of the `wanboot.conf` file. - *net-ip* is the IP address of the client's subnet. - *client-ID* can be a user-defined ID or the DHCP client ID.	`bootconfchk /etc/netboot/`*net-ip*`/`*client-ID*`/wanboot.conf`
Check for WAN boot installation support in the client OBP.	`eeprom | grep network-boot-arguments`

TABLE 46–2 Preparing the WAN Boot Security Files

Task and Description	Command
Create a master HMAC SHA1 key for the WAN boot server.	`wanbootutil keygen -m`
Create a HMAC SHA1 hashing key for the client. - *net-ip* is the IP address of the client's subnet. - *client-ID* can be a user-defined ID or the DHCP client ID.	`wanbootutil keygen -c -o net=`*net-ip*`,cid=`*client-ID*`,type=sha1`

TABLE 46–2 Preparing the WAN Boot Security Files *(Continued)*

Task and Description	Command
Create an encryption key for the client. ■ *net-ip* is the IP address of the client's subnet. ■ *client-ID* can be a user-defined ID or the DHCP client ID. ■ *key-type* is either `3des` or `aes`.	`wanbootutil keygen -c -o net=`*net-ip*`,cid=`*client-ID*`,type=`*key-type*
Split a PKCS#12 certificate file and insert the certificate in the client's `truststore`. ■ *p12cert* is the name of the PKCS#12 certificate file. ■ *net-ip* is the IP address of the client's subnet. ■ *client-ID* can be a user-defined ID or the DHCP client ID.	`wanbootutil p12split -i` *p12cert* `-t /etc/netboot/`*net-ip*`/`*client-ID*`/truststore`
Split a PKCS#12 certificate file and insert the client certificate in the client's `certstore` ■ *p12cert* is the name of the PKCS#12 certificate file. ■ *net-ip* is the IP address of the client's subnet. ■ *client-ID* can be a user-defined ID or the DHCP client ID. ■ *keyfile* is the name of the client's private key.	`wanbootutil p12split -i` *p12cert* `-c /etc/netboot/`*net-ip*`/`*client-ID*`/certstore -k` *keyfile*
Insert the client private key from a split PKCS#12 file in the client's `keystore`. ■ *keyfile* is the name of the client's private key. ■ *net-ip* is the IP address of the client's subnet. ■ *client-ID* can be a user-defined ID or a DHCP client ID.	`wanbootutil keymgmt -i -k` *keyfile* `-s /etc/netboot/`*net-ip*`/`*client-ID*`/keystore -o type=rsa`
Display the value of a HMAC SHA1 hashing key. ■ *net-ip* is the IP address of the client's subnet. ■ *client-ID* can be a user-defined ID or the DHCP client ID.	`wanbootutil keygen -d -c -o net=`*net-ip*`,cid=`*client-ID*`,type=sha1`

TABLE 46–2 Preparing the WAN Boot Security Files *(Continued)*

Task and Description	Command
Display the value of an encryption key. - *net-ip* is the IP address of the client's subnet. - *client-ID* can be a user-defined ID or the DHCP client ID. - *key-type* is either `3des` or `aes`.	`wanbootutil keygen -d -c -o net=`*net-ip*`,cid=`*client-ID*`,type=`*key-type*
Insert a hashing key or an encryption key on a running system. *key-type* can have a value of `sha1`, `3des`, or `aes`.	`/usr/lib/inet/wanboot/ickey -o type=`*key-type*

OBP Commands

The following table lists the OBP commands that you type at the client `ok` prompt to perform a WAN boot installation.

TABLE 46–3 OBP Commands for a WAN Boot Installation

Task and Description	OBP Command
Begin an unattended WAN boot installation.	`boot net – install`
Begin an interactive WAN boot installation.	`boot net –o prompt - install`
Begin a WAN boot installation from a local CD.	`boot cdrom –F wanboot - install`
Install a hashing key before you begin a WAN boot installation.*key-value* is the hexadecimal value of the hashing key.	`set-security-key wanboot-hmac-sha1` *key-value*
Install an encryption key before you begin a WAN boot installation. - *key-type* is either `wanboot-3des` or `wanboot-aes`. - *key-value* is the hexadecimal value of the encryption key.	`set-security-key` *key-type key-value*
Verify that key values are set in OBP.	`list-security-keys`

TABLE 46–3 OBP Commands for a WAN Boot Installation *(Continued)*

Task and Description	OBP Command
Set client configuration variables before you begin your WAN boot installation. ■ *client-IP* is the IP address of the client. ■ *router-ip* is the IP address of the network router. ■ *mask-value* is the subnet mask value. ■ *client-name* is the host name of the client. ■ *proxy-ip* is the IP address of the network's proxy server. ■ *wanbootCGI-path* is the path to the `wanbootCGI` programs on the web server.	`setenv network-boot-arguments host-ip=`*client-IP*`,router-ip=`*router-ip*`,subnet-mask=`*mask-value*`,hostname=`*client-name*`,http-proxy=`*proxy-ip*`,file=`*wanbootCGI-path*
Check the network device alias.	`devalias`
Set the network device alias, where *device-path* is the path to the primary network device.	■ To set the alias for the current installation only, type `devalias net` *device-path*. ■ To permanently set the alias, type `nvvalias net` *device-path*.

System Configuration File Settings and Syntax

The system configuration file enables you to direct the WAN boot installation programs to the following files.

- `sysidcfg`
- `rules.ok`
- Custom JumpStart profile

The system configuration file is a plain text file, and must be formatted in the following pattern.

setting=value

The `system.conf` file must contain the following settings.

`SsysidCF=`*sysidcfg-file-URL*
This setting points to the directory on the install server that contains the `sysidcfg` file. For WAN installations that use HTTPS, set the value to a valid HTTPS URL.

`SjumpsCF=`*jumpstart-files-URL*
This setting points to the custom JumpStart directory that contains the `rules.ok` and profile files. For WAN installations that use HTTPS, set the value to a valid HTTPS URL.

You can store the system.conf in any directory that is accessible to the WAN boot server.

wanboot.conf File Parameters and Syntax

The wanboot.conf file is a plain-text configuration file that the WAN boot installation programs use to perform a WAN installation. The following programs and files use the information included in the wanboot.conf file to install the client machine.

- wanboot-cgi program
- WAN boot file system
- WAN boot miniroot

Save the wanboot.conf file in the appropriate client subdirectory in the /etc/netboot hierarchy on the WAN boot server. For information on how to define the scope of your WAN boot installation with the /etc/netboot hierarchy, see "Creating the /etc/netboot Hierarchy on the WAN Boot Server" on page 534.

You specify information in the wanboot.conf file by listing parameters with associated values in the following format.

parameter=value

Parameter entries cannot span lines. You can include comments in the file by preceding the comments with the # character.

For detailed information about the wanboot.conf file, see the man page wanboot.conf(4).

You must set the following parameters in the wanboot.conf file.

boot_file=*wanboot-path*

This parameter specifies the path to the wanboot program. The value is a path relative to the document root directory on the WAN boot server.

```
boot_file=/wanboot/wanboot.s9_sparc
```

root_server=*wanbootCGI-URL*/wanboot-cgi

This parameter specifies the URL of the wanboot-cgi program on the WAN boot server.

- Use an HTTP URL if you are performing a WAN boot installation without client or server authentication.

  ```
  root_server=http://www.example.com/cgi-bin/wanboot-cgi
  ```

- Use an HTTPS URL if you are performing a WAN boot installation with server authentication, or server and client authentication.

  ```
  root_server=https://www.example.com/cgi-bin/wanboot-cgi
  ```

`root_file=`*miniroot-path*

This parameter specifies the path to the WAN boot miniroot on the WAN boot server. The value is a path relative to the document root directory on the WAN boot server.

```
root_file=/miniroot/miniroot.s9_sparc
```

`signature_type=sha1 |` *empty*

This parameter specifies the type of hashing key to to use to check the integrity of the data and files that are transmitted.

- For WAN boot installations that use a hashing key to protect the `wanboot` program, set this value to `sha1`.

  ```
  signature_type=sha1
  ```

- For insecure WAN installations that do not use a hashing key, leave this value blank.

  ```
  signature_type=
  ```

`encryption_type=3des | aes |` *empty*

This parameter specifies the type of encryption to use to encrypt the `wanboot` program and WAN boot file system.

- For WAN boot installations that use HTTPS, set this value to `3des` or `aes` to match the key formats you use. You must also set the `signature_type` keyword value to `sha1`.

  ```
  encryption_type=3des
  ```

  or

  ```
  encryption_type=aes
  ```

- For an insecure WAN boot installations that do not use encryption key, leave this value blank.

  ```
  encryption_type=
  ```

`server_authentication=yes | no`

This parameter specifies if the server should be authenticated during the WAN boot installation.

- For WAN boot installations with server authentication or server and client authentication, set this value to `yes`. You must also set the value of `signature_type` to `sha1`, `encryption_type` to `3des` or `aes`, and the URL of `root_server` to an HTTPS value.

  ```
  server_authentication=yes
  ```

- For insecure WAN boot installations that do not use server authentication or server and client authentication, set this value to no. You can also leave the value blank.

  ```
  server_authentication=no
  ```

client_authentication=yes | no
This parameter specifies if the client should be authenticated during a WAN boot installation.

- For WAN boot installations with server and client authentication, set this value to yes. You must also set the value of signature_type to sha1, encryption_type to 3des or aes, and the URL of root_server to an HTTPS value.

  ```
  client_authentication=yes
  ```

- For WAN boot installations that do not use client authentication, set this value to no. You can also leave the value blank.

  ```
  client_authentication=no
  ```

resolve_hosts=*hostname* | *empty*
This parameter specifies additional hosts that need to be resolved for the wanboot-cgi program during the installation.

Set the value to the host names of systems that are not specified previously in the wanboot.conf file or in a client certificate.

- If all the required hosts are listed in the wanboot.conf file or the client certificate, leave this value blank.

  ```
  resolve_hosts=
  ```

- If specific hosts are not listed in the wanboot.conf file or the client certificate, set the value to these host names.

  ```
  resolve_hosts=seahag,matters
  ```

boot_logger=*bootlog-cgi-path* | *empty*
This parameter specifies the URL to the bootlog-cgi script on the logging server.

- To record boot or installation log messages on a dedicated logging server, set the value to the URL of the bootlog-cgi script on the logging server.

  ```
  boot_logger=http://www.example.com/cgi-bin/bootlog-cgi
  ```

- To display boot and installation messages on the client console, leave this value blank.

  ```
  boot_logger=
  ```

system_conf=system.conf | *custom-system-conf*
This parameter specifies the path to the system configuration file that includes the location of sysidcfg and custom JumpStart files.

Set the value to the path to the sysidcfg and custom JumpStart files on the web server.

```
system_conf=sys.conf
```

APPENDIX **A**

Solaris Software Reference (Topics)

This section provides reference information.

Appendix B	Describes the primary DVD and CDs that are included in the media kits for Solaris 9.
Appendix C	Contains a list of the platform names and groups of various hardware platforms.
Appendix D	Contains a list of the values that are needed to set the `locale` keyword in a profile or to preconfigure a locale.
Appendix E	Contains a list of specific error messages and general problems you might encounter when you install Solaris software.
Appendix F	Describes how to use the Solaris Web Start program to install or upgrade to the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive.
Appendix G	Describes requirements that keep a custom JumpStart installation non-interactive and avoids modifying the currently running system so you can upgrade with Solaris Live Upgrade.
Appendix H	Provides instructions on checking patches before upgrading the Solaris operating environment to an update release.
Appendix I	Describes how to copy the boot image from a Solaris Device Configuration Assistant image to a diskette.
Appendix J	Describes changes to this book for Solaris 9 update releases.

APPENDIX **B**

Organization of Solaris 9 Media (Reference)

This chapter describes the primary DVD and CD media that are included in the media kits for Solaris 9 software.

SPARC: Solaris 9 Media

The following tables list the primary DVD and CD media for the Solaris 9 release. You receive a multilingual media kit that contains DVD or CD media for English, other languages, and locale software.

TABLE B–1 SPARC: Multilingual DVD Media

DVD Title	Description
Solaris *SPARC Platform Edition* DVD	Contains the following: ■ Software, tools, and configuration information to install the Solaris product and all partial locales ■ ExtraValue software that includes supported and unsupported software ■ Installers ■ Source code for some third-party public domain software ■ Interface software and documentation that have been localized ■ The Solaris documentation set for English, European, and Asian languages that includes Japanese For directory structures, see Figure B–1.

TABLE B–2 SPARC: Multilingual CD Media

CD Title	Description
Solaris Multilingual Installation *SPARC Platform Edition* CD	Contains scripts and localized installers to install Solaris software and all partial locales. For directory structures, see Figure B–2.
Solaris Software 1 of 2 *SPARC Platform Edition* CD	Contains the software, tools, and configuration information to install the Solaris product. For directory structures, see Figure B–3.
Solaris Software 2 of 2 *SPARC Platform Edition* CD	Contains the following: ■ A limited number of packages, which the software prompts you to install if necessary ■ ExtraValue software that includes supported and unsupported software ■ Installers ■ Source code for some third-party public domain software For directory structures, see Figure B–4. The installation program prompts you for this CD if necessary.
Solaris *SPARC Platform Edition* Languages CD	Contains interface software and documentation that has been localized. For directory structures, see Figure B–5. The installation program prompts you for this CD if necessary to support languages for specific geographic regions.
Solaris Documentation 1 of 2 CD	Contains the Solaris documentation set for English and European languages.
Solaris Documentation 2 of 2 CD	Contains the Solaris documentation set for Asian languages that includes Japanese.

SPARC: Directory Organization of Solaris 9 Media

This section describes top-level directories on each DVD and CD.

SPARC: Solaris *SPARC Platform Edition* DVD Directory Structure

The following figure shows the directory structure on the Solaris *SPARC Platform Edition* DVD.

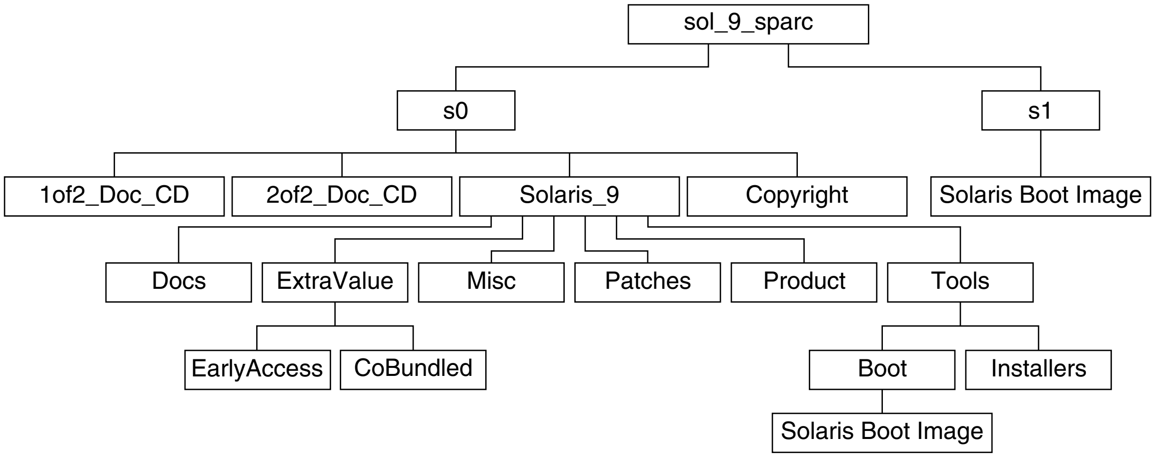


FIGURE B–1 Solaris *SPARC Platform Edition* DVD

- Slice 0 (s0) contains the `Copyright` and `Solaris_9` directories. The `Solaris_9` directory contains all the tools, software, and configuration information necessary to install the Solaris 9 software product. The `Solaris_9` directory contains the following directories:
  - `ExtraValue` – Two subdirectories:
    - `CoBundled` – Fully supported Solaris products not directly part of the Solaris operating environment that includes Solaris Web Start Wizards SDK 3.0.1 and SunScreen 3.2
    - `EarlyAccess` – Preliminary evaluation software
  - `Misc` – The `jumpstart_sample` directory, which includes a `rules` file, a `check` script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
  - `Patches` – All the Solaris 9 patches available at the time of this release.
  - `Product` – The Solaris 9 packages and control files.
  - `Tools` – The Solaris 9 installation tools, which include the following:
    - The `Boot` subdirectory, which contains the Solaris 9 miniroot.
    - The scripts `add_install_client`, `dial`, `rm_install_client`, and `setup_install_server`.

    - An Installers subdirectory that contains installers and supporting Java™ class files to install Solaris Live Upgrade and the Solaris Software 2 of 2 *SPARC Platform Edition* CD. Solaris Live Upgrade is automatically installed when you install the Solaris software. However, in order to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see "To Install Solaris Live Upgrade With the Solaris Web Start Installer" on page 412.
  - 1of2_Doc_CD – Documentation for English and European Languages
  - 2of2_Doc_CD – Documentation for Asian Languages
- Slice 1 (s1) contains the Solaris 9 miniroot.

SPARC: Solaris Installation CD Directory Structure

The following figure shows the directory structure on the Solaris Multilingual Installation *SPARC Platform Edition* CD or Solaris *SPARC Platform Edition* Installation CD.

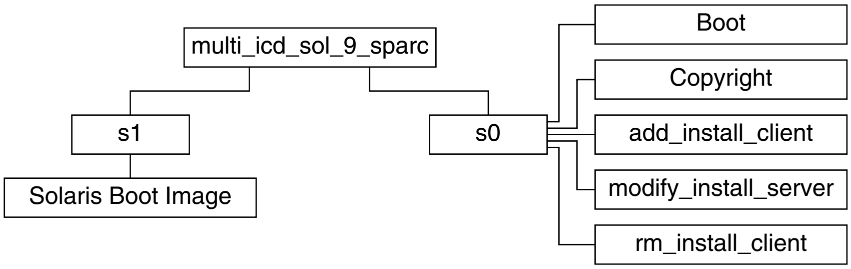


FIGURE B–2 SPARC: Solaris *SPARC Platform Edition* Installation CD

The directory multi_icd_sol_sparc contains scripts that install the Solaris software and includes the Boot subdirectory for the Solaris 9 miniroot. The directories and scripts include the following:

- Slice 0 (s0) contains scripts that install the Solaris software and a Boot subdirectory that contains the Solaris miniroot. These scripts include the following:
  - add_install_client
  - modify_install_server
  - rm_install_client
- Slice 1 (s1) contains the Solaris 9 miniroot.

SPARC: Solaris Software *SPARC Platform Edition* CDs Directory Structures

The following figures show the directory structure on the Solaris Software *SPARC Platform Edition* CDs.

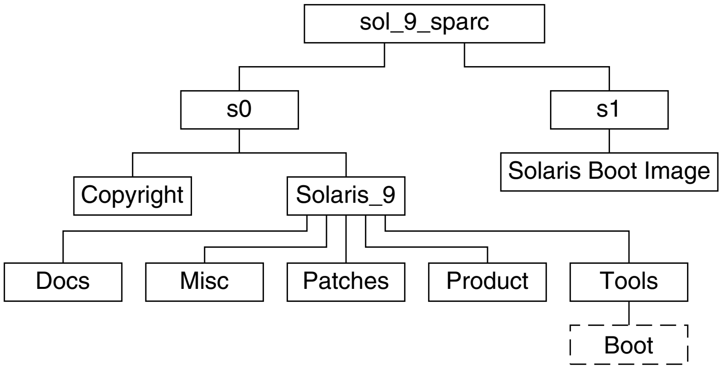


FIGURE B–3 SPARC: Solaris Software 1 of 2 *SPARC Platform Edition* CD

- Slice 0 (s0) contains the `Copyright` and `Solaris_9` directories. The `Solaris_9` directory contains all the tools, software, and configuration information necessary to install, at a minimum, the Solaris 9 software product, including the Core System Support and End User System Support software groups. The `Solaris_9` directory contains the following directories:
  - `Docs` – An empty directory.
  - `Misc` – The `jumpstart_sample` directory, which includes a `rules` file, a `check` script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
  - `Patches` – All the Solaris 9 patches available at the time the Solaris Software 1 of 2 *SPARC Platform Edition* CD was created.
  - `Product` – The Solaris 9 packages and control files.
  - `Tools` – The Solaris 9 installation tools, which include a symbolic link to the `boot` miniroot on slice 1, `add_install_client`, `dial`, `rm_install_client`, and `setup_install_server`.
- Slice 1 (s1) contains the Solaris 9 miniroot.

The following figure shows the directories on the Solaris Software 2 of 2 *SPARC Platform Edition* CD.

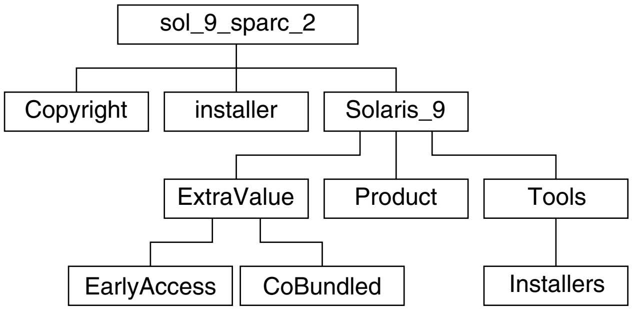


FIGURE B–4 SPARC: Solaris Software 2 of 2 *SPARC Platform Edition* CD

The `sol_sparc_2` directory contains `Copyright`, Solaris Web Start `installer`, and `Solaris_9` directories. The `Solaris_9` directory contains the following subdirectories.

- `ExtraValue` – Two subdirectories:
  - `CoBundled` – Fully supported Solaris products not directly part of the Solaris operating environment
  - `EarlyAccess` – Preliminary evaluation software
- `Product` – A limited number of packages for the Developer System Support, Entire Distribution, and Entire Distribution Plus OEM Support software groups
- `Tools` – Solaris 9 installation tools that include the following:
  - The `add_to_install_server` script
  - An `Installers` subdirectory that contains installers and supporting Java class files to install Solaris Live Upgrade and the Solaris Software 2 of 2 *SPARC Platform Edition* CD. Solaris Live Upgrade is automatically installed when you install the Solaris software, but to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see "To Install Solaris Live Upgrade With the Solaris Web Start Installer" on page 412.

SPARC: Solaris *SPARC Platform Edition* Languages CD Directory Structure

The following figure shows the directory structure of the Solaris *SPARC Platform Edition* Languages CD.

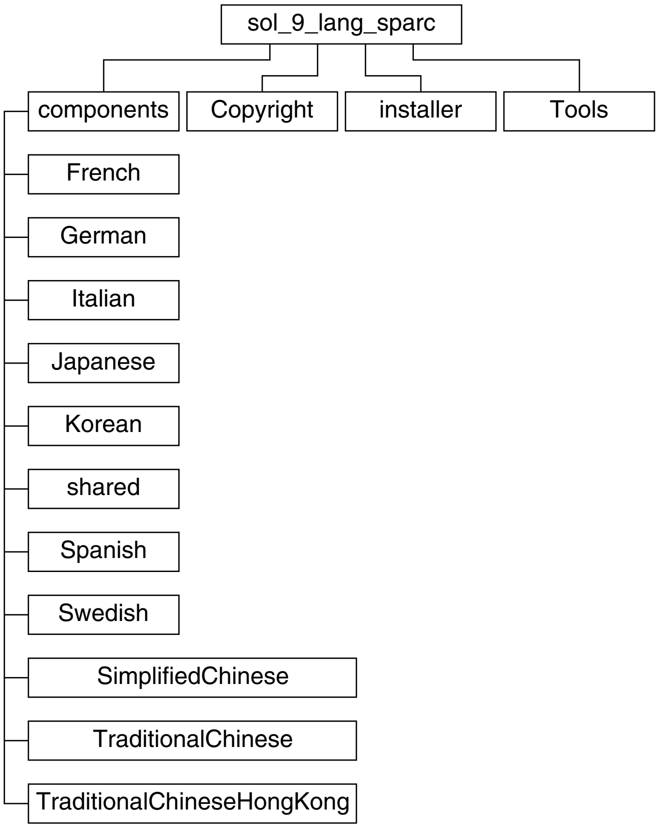


FIGURE B–5 Solaris *SPARC Platform Edition* Languages CD

The `sol_lang_sparc` directory contains the Solaris Web Start installer, which installs the Solaris language and locale software. The `sol_lang_sparc` directory contains these directories:

- `components` – SimplifiedChinese, French, German, Italian, Japanese, Korean, Spanish, Swedish, TraditionalChinese, and TraditionalChineseHongKong locale packages and packages that are shared by all locales
- `Copyright` – Copyright page
- `installer` – Solaris Web Start installer
- `Tools` – The `add_to_install_server` script for creating a combined installation image

For more information on locale software, see *International Language Environments Guide*.

x86: Solaris 9 Media

The following tables list the primary DVD and CD media for Solaris 9 *x86 Platform Edition*. You receive a multilingual media kit that contains DVD or CD media for English, other languages, and locale software.

TABLE B–3 x86: Multilingual DVD Media

DVD Title	Description
Solaris *x86 Platform Edition* DVD	Contains the following: ■ Software, tools, and configuration information to install the Solaris product and all partial locales. This includes the Solaris Device Configuration Assistant, which performs various configuration and booting tasks. ■ ExtraValue software, which includes supported and unsupported software. ■ Installers. ■ Source code for some third-party public domain software. ■ Interface software and documentation that have been localized. ■ The Solaris documentation set for English, European, and Asian languages that includes Japanese. For directory structures, see Figure B–6.

TABLE B–4 x86: Multilingual CD Media

CD Title	Description
Solaris Installation Multilingual *x86 Platform Edition* CD	Contains scripts and localized installers to install Solaris software and all partial locales. This includes the Solaris Device Configuration Assistant, which performs various configuration and booting tasks. For directory structures, see Figure B–7.
Solaris Software 1 of 2 *x86 Platform Edition* CD	Contains the software, tools, and configuration information to install the Solaris product. This includes the Solaris Device Configuration Assistant, which performs various configuration and booting tasks. For directory structures, see Figure B–8.

TABLE B–4 x86: Multilingual CD Media *(Continued)*

CD Title	Description
Solaris Software 2 of 2 *x86 Platform Edition* CD	Contains the following: ■ A limited number of packages, which the software prompts you to install if necessary ■ ExtraValue software that includes supported and unsupported software ■ Installers ■ Source code for some third-party public domain software For directory structures, see Figure B–9. The installation program prompts you for this CD if necessary.
Solaris *x86 Platform Edition* Languages CD	Contains interface software and documentation that are localized. For directory structures, see Figure B–10. The installation program prompts you for this CD if necessary to support languages for specific geographic regions.
Solaris Documentation 1 of 2 CD	Contains the Solaris documentation set for English and European languages.
Solaris Documentation 2 of 2 CD	Contains the Solaris documentation set for Asian languages that includes Japanese.

x86: Directory Organization of Solaris 9 Media

This section describes top-level directories on each DVD and CD.

Solaris *x86 Platform Edition* DVD Directory Structure

The following figure shows the directories on the Solaris *x86 Platform Edition* DVD.

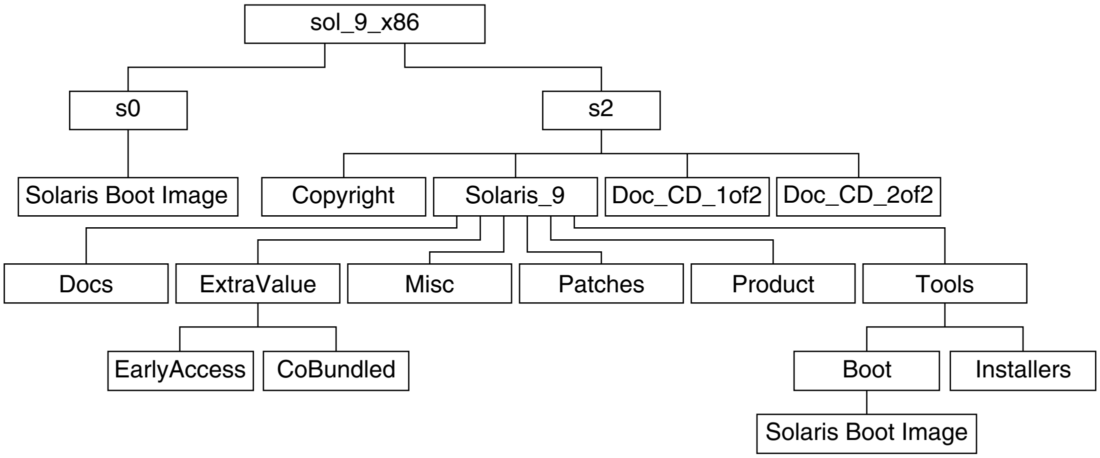


FIGURE B–6 Solaris *x86 Platform Edition* DVD

- Slice 0 (s0) contains the Solaris 9 miniroot.
- Slice 2 (s2) contains the `Copyright` and `Solaris_9` directories. The `Solaris_9` directory contains all the tools, software, and configuration information necessary to install the Solaris 9 software product. Slice 2 contains the following directories:
  - `ExtraValue` – Two subdirectories:
    - `CoBundled` – Fully supported Solaris products not directly part of the Solaris operating environment that includes Solaris Web Start Wizards SDK 3.0.1 and SunScreen 3.2
    - `EarlyAccess` – Preliminary evaluation software
  - `Misc` – The `jumpstart_sample` directory, which includes a `rules` file, a `check` script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
  - `Patches` – All the Solaris 9 patches available at the time of this release.
  - `Product` – The Solaris 9 packages and control files.
  - `Tools` – The Solaris 9 installation tools, which include the following.
    - The `Boot` subdirectory that contains the Solaris 9 miniroot.
    - The scripts `add_install_client`, `dial`, `rm_install_client`, and `setup_install_server`
    - An `Installers` subdirectory that contains installers and supporting Java class files to install Solaris Live Upgrade and the Solaris Software 2 of 2 *x86 Platform Edition* CD. Solaris Live Upgrade is automatically installed when you install the Solaris software. However, in order to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see “To Install Solaris Live Upgrade With the Solaris Web Start

Installer" on page 412.

  - A `d1_image` subdirectory that contains the Solaris Device Configuration Assistant software that can be copied to a diskette.

- `1of2_Doc_CD` – Documentation for English and European Languages
- `2of2_Doc_CD` – Documentation for Asian Languages

x86: Solaris Installation CD Directory Structure

The following figure shows the directories on the Solaris Installation Multilingual *x86 Platform Edition* CD or Solaris *x86 Platform Edition* Installation CD.

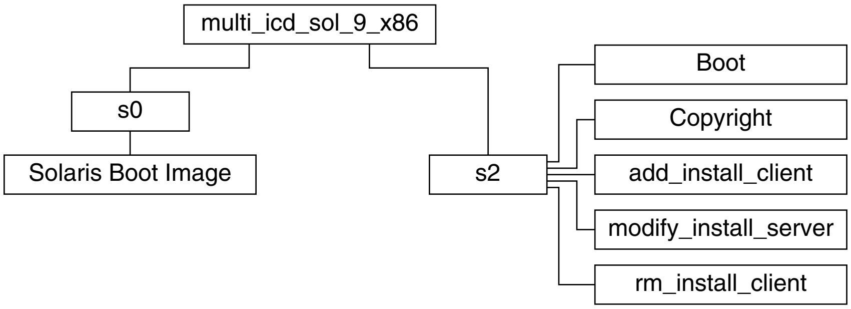


FIGURE B–7 x86: Solaris *x86 Platform Edition* Installation CD

The directory `multi_icd_sol_x86` contains scripts that install the Solaris software and includes the `Boot` subdirectory for the Solaris 9 miniroot. The directories and scripts include the following:

- Slice 0 (s0) contains the Solaris 9 miniroot.
- Slice 2 (s2) contains scripts that install the Solaris software and a Boot subdirectory that contains the Solaris miniroot. These scripts include the following:
  - `add_install_client`
  - `modify_install_server`
  - `rm_install_client`

x86: Solaris Software *x86 Platform Edition* CDs Directory Structures

The following figures show the directory structure of the Solaris Software *x86 Platform Edition* CDs.

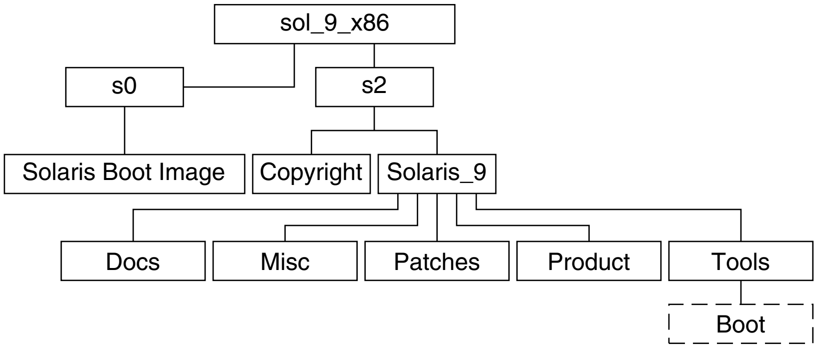


FIGURE B–8 x86: Solaris Software 1 of 2 *x86 Platform Edition* CD

- Slice 0 (s0) contains the Solaris 9 miniroot.
- Slice 2 (s2) contains the `Copyright` and `Solaris_9` directories. The `Solaris_9` directory contains all the tools, software, and configuration necessary to install, at a minimum, the Solaris 9 software product, including the Core System Support and End User System Support software groups. The `Solaris_9` directory contains the following directories:
  - `Docs` – Empty directory.
  - `Misc` – The `jumpstart_sample` directory, which includes a `rules` file, a `check` script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
  - `Patches` – All the Solaris 9 patches available at the time the Solaris Software 1 of 2 *x86 Platform Edition* CD was created.
  - `Product` – The Solaris 9 packages and control files.
  - `Tools` – The Solaris 9 installation tools, which include a symbolic link to the `boot` miniroot on slice 0, `add_install_client`, `dial`, `rm_install_client`, and `setup_install_server`.

The following figure shows the directories on the Solaris Software 2 of 2 *x86 Platform Edition* CD.

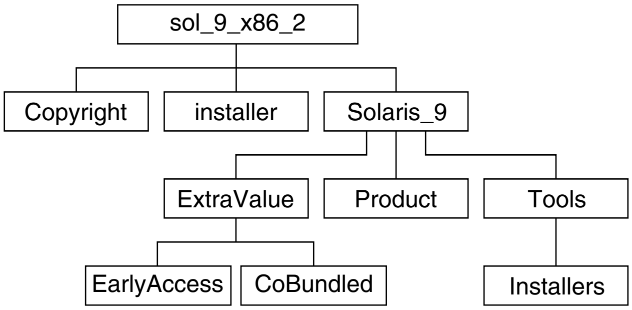


FIGURE B–9 Solaris Software 2 of 2 *x86 Platform Edition* CD

The sol_x86_2 directory contains Copyright, Solaris Web Start installer, and Solaris_9 directories. The Solaris_9 directory contains the following:

- ExtraValue – Two subdirectories:
  - CoBundled – Fully supported Solaris products not directly part of the Solaris operating environment
  - EarlyAccess – Preliminary evaluation software
- Product – A limited number of packages, Developer System Support, Entire Distribution, and Entire Distribution Plus OEM Support software groups.
- Tools – This directory includes the following:
  - The add_to_install_server script.
  - An Installers subdirectory that contains an installer and supporting Java class files to install Solaris Live Upgrade and the Solaris Software 2 of 2 *x86 Platform Edition* CD. Solaris Live Upgrade is automatically installed when you install the Solaris software. However, in order to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see "To Install Solaris Live Upgrade With the Solaris Web Start Installer" on page 412.
  - A d1_image subdirectory that contains the Solaris Device Configuration Assistant software that can be copied to a diskette.

x86: Solaris *x86 Platform Edition* Languages CD Directory Structures

The following figure shows the directory structure of the Solaris *x86 Platform Edition* Languages CD.

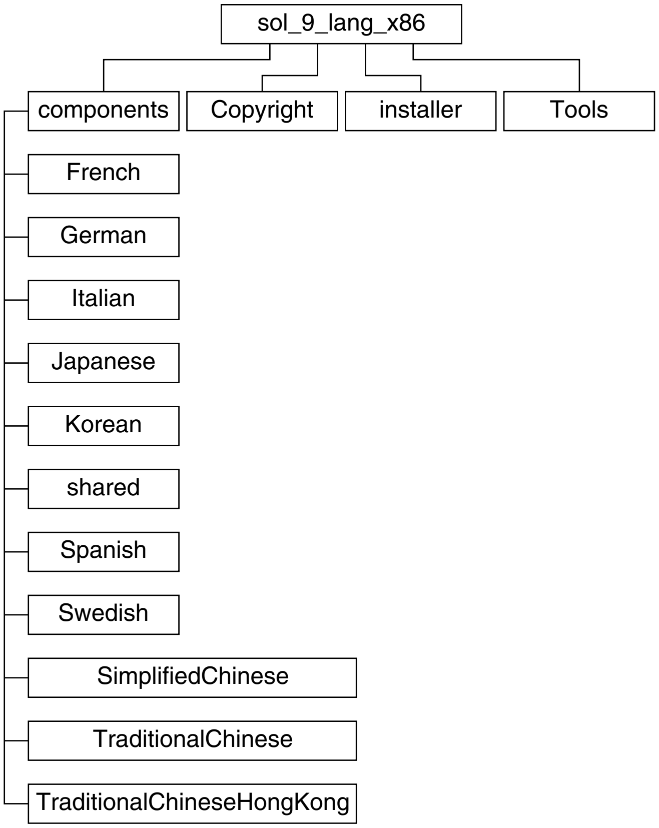


FIGURE B–10 Solaris *x86 Platform Edition* Languages CD

The `sol_lang_x86` directory contains the Solaris Web Start program, which installs the Solaris language and locale software. You can choose to install the nine languages in default directories or install specific languages. The `sol_lang_x86` directory contains these directories:

- `Components` – Contains SimplifiedChinese, French, German, Italian, Japanese, Korean, Spanish, Swedish, TraditionalChinese, and TraditionalChineseHongKong locale packages and also includes packages that are shared by all locales
- `Copyright` – The copyright page
- `installer` – The Web Start installer
- `Tools` – The `add_to_install_server` script for creating an install server

For more information on locale software, see *International Language Environments Guide*.

APPENDIX **C**

Platform Names and Groups (Reference)

You need to know your system architecture (platform group) if you are adding clients for a network installation and the platform name if you are writing a custom JumpStart installation rules file.

Some examples of platform names and groups follow. For a full list of SPARC systems, see *Solaris 9 Sun Hardware Platform Guide*.

TABLE C–1 Example of Platform Names and Groups

System	Platform Name	Platform Group
Ultra 5	SUNW,Ultra-5_10	sun4u
x86 based	i86pc	i86pc

Note – On a running system, you can also use the `uname -i` command to determine a system's *platform name* or the `uname -m` command to determine a system's *platform group*.

APPENDIX D

Locale Values (Reference)

The tables below list the values that are needed to set the `locale` keyword in a profile or to preconfigure a locale.

A *locale* determines how online information is displayed in a specific language and region. A language might also include more than one locale to accommodate regional differences, such as differences in the format of date and time, numeric and monetary conventions, and spelling. For additional information about locales, see the *International Language Environments Guide*.

TABLE D–1 Asia

Locale	User Interface	Territory	Codeset	Language Support
`hi_IN.UTF-8`	English	India	UTF-8[1]	Hindi (UTF-8) Unicode 3.1
`ja`	Japanese	Japan	eucJP[2]	Japanese (EUC)
				JIS X 0201-1976
				JIS X 0208-1990
				JIS X 0212-1990
`ja_JP.eucJP`	Japanese	Japan	eucJP	Japanese (EUC)
				JIS X 0201-1976
				JIS X 0208-1990
				JIS X 0212-1990
`ja_JP.PCK`	Japanese	Japan	PCK[3]	Japanese (PC kanji)
				JIS X 0201-1976

[1] UTF-8 is the UTF-8 defined in ISO/IEC 10646–1:2000 and also Unicode 3.1.

[2] eucJP signifies the Japanese EUC codeset. Specification of `ja_JP.eucJP` locale conforms to UI_OSF Japanese Environment Implementation Agreement Version 1.1 and `ja` locale conforms to the traditional specification from the past Solaris releases.

[3] PCK is also known as Shift_JIS (SJIS).

TABLE D–1 Asia *(Continued)*

Locale	User Interface	Territory	Codeset	Language Support
				JIS X 0208-1990
`ja_JP.UTF-8`	Japanese	Japan	UTF-8	Japanese (UTF-8) Unicode 3.1
`ko_KR.EUC`	Korean	Korea	1001	Korean (EUC) KS X 1001
`ko_KR.UTF-8`	Korean	Korea	UTF-8	Korean (UTF-8) Unicode 3.1
`th_TH.UTF-8`	English	Thailand	UTF-8	Thai (UTF-8) Unicode 3.1
`th_TH.TIS620`	English	Thailand	TIS620.2533	Thai TIS620.2533
`zh_CN.EUC`	Simplified Chinese	PRC	gb2312[4]	Simplified Chinese (EUC) GB2312-1980
`zh_CN.GBK`	Simplified Chinese	PRC	GBK[5]	Simplified Chinese (GBK)
`zh_CN.GB18030`	Simplified Chinese	PRC	GB18030–2000	Simplified Chinese (GB18030–2000) GB18030–2000
`zh_CN.UTF-8`	Simplified Chinese	PRC	UTF-8	Simplified Chinese (UTF-8) Unicode 3.1
`zh_HK.BIG5HK`	Traditional Chinese	Hong Kong	Big5+HKSCS	Traditional Chinese (BIG5+HKSCS)
`zh_HK.UTF-8`	Traditional Chinese	Hong Kong	UTF-8	Traditional Chinese (UTF-8) Unicode 3.1
`zh_TW.EUC`	Traditional Chinese	Taiwan	cns11643	Traditional Chinese (EUC) CNS 11643-1992
`zh_TW.BIG5`	Traditional Chinese	Taiwan	BIG5	Traditional Chinese (BIG5)
`zh_TW.UTF-8`	Traditional Chinese	Taiwan	UTF-8	Traditional Chinese (UTF-8) Unicode 3.1

[4] gb2312 signifies Simplified Chinese EUC codeset, which contains GB 1988–80 and GB 2312–80.

[5] GBK signifies GB extensions. This includes all GB 2312–80 characters and all Unified Han characters of ISO/IEC 10646–1, as well as Japanese Hiragana and Katakana characters. It also includes many characters of Chinese, Japanese, and Korean character sets and of ISO/IEC 10646–1.

TABLE D–2 Australasia

Locale	User Interface	Territory	Codeset	Language Support
`en_AU.ISO8859-1`	English	Australia	ISO8859-1	English (Australia)
`en_NZ.ISO8859-1`	English	New Zealand	ISO8859-1	English (New Zealand)

TABLE D–3 Central America

Locale	User Interface	Territory	Codeset	Language Support
`es_CR.ISO8859-1`	Spanish	Costa Rica	ISO8859-1	Spanish (Costa Rica)
`es_GT.ISO8859-1`	Spanish	Guatemala	ISO8859-1	Spanish (Guatemala)
`es_NI.ISO8859-1`	Spanish	Nicaragua	ISO8859-1	Spanish (Nicaragua)
`es_PA.ISO8859-1`	Spanish	Panama	ISO8859-1	Spanish (Panama)
`es_SV.ISO8859-1`	Spanish	El Salvador	ISO8859-1	Spanish (El Salvador)

TABLE D–4 Central Europe

Locale	User Interface	Territory	Codeset	Language Support
`cs_CZ.ISO8859-2`	English	Czech Republic	ISO8859-2	Czech (Czech Republic)
`de_AT.ISO8859-1`	German	Austria	ISO8859-1	German (Austria)
`de_AT.ISO8859-15`	German	Austria	ISO8859-15	German (Austria, ISO8859-15 - Euro)
`de_CH.ISO8859-1`	German	Switzerland	ISO8859-1	German (Switzerland)
`de_DE.UTF-8`	German	Germany	UTF-8	German (Germany, Unicode 3.1)
`de_DE.ISO8859-1`	German	Germany	ISO8859-1	German (Germany)
`de_DE.ISO8859-15`	German	Germany	ISO8859-15	German (Germany, ISO8859-15 - Euro)
`fr_CH.ISO8859-1`	French	Switzerland	ISO8859-1	French (Switzerland)
`hu_HU.ISO8859-2`	English	Hungary	ISO8859-2	Hungarian (Hungary)
`pl_PL.ISO8859-2`	English	Poland	ISO8859-2	Polish (Poland)
`pl_PL.UTF-8`	English	Poland	UTF-8	Polish (Poland, Unicode 3.1)
`sk_SK.ISO8859-2`	English	Slovakia	ISO8859-2	Slovak (Slovakia)

TABLE D–5 Eastern Europe

Locale	User Interface	Territory	Codeset	Language Support
`bg_BG.ISO8859-5`	English	Bulgaria	ISO8859-5	Bulgarian (Bulgaria)
`et_EE.ISO8859-15`	English	Estonia	ISO8859-15	Estonian (Estonia)
`hr_HR.ISO8859-2`	English	Croatia	ISO8859-2	Croatian (Croatia)

TABLE D–5 Eastern Europe *(Continued)*

Locale	User Interface	Territory	Codeset	Language Support
lt_LT.ISO8859-13	English	Lithuania	ISO8859-13	Lithuanian (Lithuania)
lv_LV.ISO8859-13	English	Latvia	ISO8859-13	Latvian (Latvia)
mk_MK.ISO8859-5	English	Macedonia	ISO8859-5	Macedonian (Macedonia)
ro_RO.ISO8859-2	English	Romania	ISO8859-2	Romanian (Romania)
ru_RU.KOI8-R	English	Russia	KOI8-R	Russian (Russia, KOI8-R)
ru_RU.ANSI1251	English	Russia	ansi-1251	Russian (Russia, ANSI 1251)
ru_RU.ISO8859-5	English	Russia	ISO8859-5	Russian (Russia)
ru_RU.UTF-8	English	Russia	UTF-8	Russian (Russia, Unicode 3.1)
sh_BA.ISO8859-2@bosnia	English	Bosnia	ISO8859-2	Bosnian (Bosnia)
sl_SI.ISO8859-2	English	Slovenia	ISO8859-2	Slovenian (Slovenia)
sq_AL.ISO8859-2	English	Albania	ISO8859-2	Albanian (Albania)
sr_YU.ISO8859-5	English	Serbia	ISO8859-5	Serbian (Serbia)
tr_TR.ISO8859-9	English	Turkey	ISO8859-9	Turkish (Turkey)
tr_TR.UTF-8	English	Turkey	UTF-8	Turkish (Turkey, Unicode 3.1)

TABLE D–6 Middle East

Locale	User Interface	Territory	Codeset	Language Support
He	English	Israel	ISO8859-8	Hebrew (Israel)

TABLE D–7 North Africa

Locale	User Interface	Territory	Codeset	Language Support
ar_EG.UTF-8	English	Egypt	UTF-8	Arabic (Egypt)
Ar	English	Egypt	ISO8859-6	Arabic (Egypt)

TABLE D–8 North America

Locale	User Interface	Territory	Codeset	Language Support
`en_CA.ISO8859-1`	English	Canada	ISO8859-1	English (Canada)
`en_US.ISO8859-1`	English	USA	ISO8859-1	English (U.S.A.)
`en_US.ISO8859-15`	English	USA	ISO8859-15	English (U.S.A., ISO8859-15 - Euro)
`en_US.UTF-8`	English	USA	UTF-8	English (U.S.A., Unicode 3.1)
`fr_CA.ISO8859-1`	French	Canada	ISO8859-1	French (Canada)
`es_MX.ISO8859-1`	Spanish	Mexico	ISO8859–1	Spanish (Mexico)

TABLE D–9 Northern Europe

Locale	User Interface	Territory	Codeset	Language Support
`da_DK.ISO8859-1`	English	Denmark	ISO8859–1	Danish (Denmark)
`da_DK.ISO8859-15`	English	Denmark	ISO8859–15	Danish (Denmark, ISO8859–15–Euro)
`fi_FI.ISO8859-1`	English	Finland	ISO8859–1	Finnish (Unicode 3.1)
`fi_FI.ISO8859-15`	English	Finland	ISO8859–15	Finnish (Finland, ISO8859–15–Euro)
`fi_FI.UTF-8`	English	Finland	UTF-8	Finnish (Finland)
`is_IS.ISO8859-1`	English	Iceland	ISO8859–1	Icelandic (Iceland)
`no_NO.ISO8859-1@bokmal`	English	Norway	ISO8859–1	Norwegian (Norway-Bokmal)
`no_NO.ISO8859-1@nyorsk`	English	Norway	ISO8859–1	Norwegian (Norway-Nynorsk)
`sv_SE.ISO8859-1`	Swedish	Sweden	ISO8859–1	Swedish (Sweden)
`sv_SE.ISO8859-15`	Swedish	Sweden	ISO8859–15	Swedish (Sweden, ISO8859–15–Euro)
`sv_SE.UTF-8`	Swedish	Sweden	UTF-8	Swedish (Sweden, Unicode 3.1)

TABLE D–10 South America

Locale	User Interface	Territory	Codeset	Language Support
`es_AR.ISO8859-1`	Spanish	Argentina	ISO8859-1	Spanish (Argentina)
`es_BO.ISO8859-1`	Spanish	Bolivia	ISO8859-1	Spanish (Bolivia)
`es_CL.ISO8859-1`	Spanish	Chile	ISO8859-1	Spanish (Chile)

TABLE D–10 South America *(Continued)*

Locale	User Interface	Territory	Codeset	Language Support
`es_CO.ISO8859-1`	Spanish	Colombia	ISO8859-1	Spanish (Colombia)
`es_EC.ISO8859-1`	Spanish	Ecuador	ISO8859-1	Spanish (Ecuador)
`es_PE.ISO8859-1`	Spanish	Peru	ISO8859-1	Spanish (Peru)
`es_PY.ISO8859-1`	Spanish	Paraguay	ISO8859-1	Spanish (Paraguay)
`es_UY.ISO8859-1`	Spanish	Uruguay	ISO8859-1	Spanish (Uruguay)
`es_VE.ISO8859-1`	Spanish	Venezuela	ISO8859-1	Spanish (Venezuela)
`pt_BR.ISO8859-1`	English	Brazil	ISO8859-1	Portuguese (Brazil)
`pt_BR.UTF-8`	English	Brazil	UTF-8	Portuguese (Brazil, Unicode 3.1)

TABLE D–11 South Europe

Locale	User Interface	Territory	Codeset	Language Support
`ca_ES.ISO8859-1`	English	Spain	ISO8859-1	Catalan (Spain)
`ca_ES.ISO8859-15`	English	Spain	ISO8859-15	Catalan (Spain, ISO8859-15 - Euro)
`el_GR.ISO8859-7`	English	Greece	ISO8859-7	Greek (Greece)
`es_ES.ISO8859-1`	Spanish	Spain	ISO8859-1	Spanish (Spain)
`es_ES.ISO8859-15`	Spanish	Spain	ISO8859-15	Spanish (Spain, ISO8859-15 - Euro)
`es_ES.UTF-8`	Spanish	Spain	UTF-8	Spanish (Spain, Unicode 3.1)
`it_IT.ISO8859-1`	Italian	Italy	ISO8859-1	Italian (Italy)
`it_IT.ISO8859-15`	Italian	Italy	ISO8859-15	Italian (Italy, ISO8859-15 - Euro)
`it_IT.UTF-8`	Italian	Italy	UTF-8	Italian (Italy, Unicode 3.1)
`pt_PT.ISO8859-1`	English	Portugal	ISO8859-1	Portuguese (Portugal)
`pt_PT.ISO8859-15`	English	Portugal	ISO8859-15	Portuguese (Portugal, ISO8859-15 - Euro)

TABLE D–12 Western Europe

Locale	User Interface	Territory	Codeset	Language Support
`en_GB.ISO8859-1`	English	Great Britain	ISO8859-1	English (Great Britain)

TABLE D–12 Western Europe *(Continued)*

Locale	User Interface	Territory	Codeset	Language Support
en_IE.ISO8859-1	English	Ireland	ISO8859-1	English (Ireland)
fr_BE.ISO8859-1	French	Belgium-Walloon	ISO8859-1	French (Belgium-Walloon, Unicode 3.1)
fr_BE.UTF-8	French	Belgium-Walloon	UTF-8	French (Belgium-Walloon, Unicode 3.1)
fr_FR.ISO8859-1	French	France	ISO8859-1	French (France)
fr_FR.UTF-8	French	France	UTF-8	French (France, Unicode 3.1)
nl_BE.ISO8859-1	English	Belgium-Flemish	ISO8859-1	Dutch (Belgium-Flemish)
nl_NL.ISO8859-1	English	Netherlands	ISO8859-1	Dutch (Netherlands)

APPENDIX **E**

Troubleshooting (Tasks)

This chapter contains a list of specific error messages and general problems you might encounter when installing Solaris 9 software. The chapter also explains how to fix the problems. Start by using this list of sections to determine where in the installation process the problem occurred.

- “Problems Setting Up Network Installations” on page 631
- “Problems With Booting a System” on page 632
- “Initial Installation of the Solaris 9 Operating Environment” on page 639
- “Upgrading the Solaris 9 Operating Environment” on page 641

Note – When you see the phrase “bootable media,” this means one of the installation programs: Solaris `suninstall` program, Solaris Web Start program, or custom JumpStart.

Problems Setting Up Network Installations

`Unknown client “`*host_name*`”`

Cause: The *host_name* argument in the `add_install_client` command is not a host in the name service.

Add the host *host_name* to the name service and execute the `add_install_client` command again.

Problems With Booting a System

Booting From Media, Error Messages

`le0: No carrier - transceiver cable problem`

Cause: The system is not connected to the network.

Solution: If this is a non-networked system, ignore this message. If this is a networked system, ensure that the Ethernet cabling is attached securely.

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

Cause: The system cannot find the proper media for booting.

Solution: Verify that the system has been set up properly to install the Solaris 9 software from the network from an install server. For example, ensure that you specified the correct platform group for the system when you set it up.

Or, if you did not copy the images of the Solaris DVD or Solaris Software 1 of 2, Solaris Software 2 of 2, and Solaris Languages CDs to the install server, ensure the Solaris DVD or Solaris Software 1 of 2 CD is mounted and accessible on the install server.

`boot: cannot open /kernel/unix`(*SPARC based systems only*)

Cause: This error occurs when you override the location of the `boot - file` by explicitly setting it to `/kernel/unix`.

Solution:

- Reset the `boot -file` in the PROM to " " (blank).
- Ensure that the diag-switch is set to off and to true.

`Can't boot from file/device`

Cause: The installation media cannot find the bootable media.

Solution: Ensure that the following conditions are met:

- The DVD-ROM or CD-ROM drive is installed properly and turned on.
- Solaris DVD or the Solaris Software 1 of 2 CD is inserted into the drive.
- The disc is free of damage or dirt.

`WARNING: clock gained` *xxx* `days -- CHECK AND RESET DATE!` (*SPARC based systems only*)

Description: This is an informational message.

Solution: Ignore the message and continue with the installation.

`Not a UFS file system` (*x86 based systems only*)

Cause: When Solaris 9 software was installed (either through the Solaris `suninstall` program or custom JumpStart), no boot disk was selected. You now must use the Solaris Device Configuration Assistant *x86 Platform Edition* diskette or edit the BIOS to boot the system.

Solution:

- Insert the Solaris Device Configuration Assistant *x86 Platform Edition* diskette into the system's boot diskette drive (usually drive A). For information on accessing the Solaris Device Configuration Assistant *x86 Platform Edition* diskette, see "x86: Accessing the Solaris 9 Device Configuration Assistant and PXE" on page 32.
- If you cannot use the bootable media, go into the BIOS and select the BIOS to boot. See your BIOS documentation for instructions.

`The Solaris Installer could not find a disk that meets the criteria found in the Install documentation. Please see the documentation for more info.` (*x86 based systems only*)

Cause: You've tried to boot from the Solaris *x86 Platform Edition* Installation CD. The system does not support logical block addressing (LBA) and the Solaris Installation CD cannot be used.

Solution: Use a net image of the CD, a DVD, or the Solaris Software 1 of 2 *x86 Platform Edition* CD to install.

Booting From Media, General Problems

`The system does not boot.`

Description: When initially setting up a custom JumpStart server, you might encounter boot problems that do not return an error message. To verify information about the system and how the system is booting, run the boot command with the `-v` option. When you use the `-v` option, the boot command displays verbose debugging information on the screen.

Note – If this flag is not given, the messages are still printed, but the output is directed to the system logfile. For more information, see syslogd(1M).

Solution: For SPARC based systems, at the `ok` prompt, type the following command.

```
ok boot net -v - install
```

For x86 based systems, when the installation program prompts you to "Select type of installation," type the following command.

```
b - -v install
```

`Boot from DVD media fails on systems with Toshiba SD—M 1401 DVD-ROM`

If your system has a Toshiba SD-M1401 DVD-ROM with firmware revision 1007, the system cannot boot from the Solaris DVD.

Solution: Apply patch 111649–03, or later version, to update the Toshiba SD-M1401 DVD-ROM drive's firmware. Patch 111649–03 is included on the Solaris 9 Supplement CD.

`The installation fails after booting.` (*x86 based systems only*)

Cause: If you are installing from the Solaris Installation CD, the Solaris 9 root slice must be located within the first 1024 cylinders of the disk.

Solution: The BIOS and SCSI driver for the default boot disk must support logical block addressing (LBA). LBA enables the machine to boot beyond the 1024–cylinder limit and across Solaris disk slices. To determine if your system supports LBA, see Table 5–2. If your system does not support LBA, boot from a net image rather than the CD.

`The system hangs or panics when non-memory PC cards are inserted.` (*x86 based systems only*)

Cause: Non-memory PC cards cannot use the same memory resources used by other devices.

Solution: To correct this problem, see the instructions for your PC card and check for the address range.

`The IDE BIOS primary drive on your system was not detected by the Solaris Device Configuration Assistant` *x86 Platform Edition* `diskette during the pre-booting phase.` (*x86 based systems only*)

Solution:

- If you are using old drives, they might be unsupported. Check your hardware manufacturer's documentation.
- Make sure the ribbon and power cables are plugged in correctly. Check the manufacturer's documentation.
- If only one drive is attached to the controller, designate the drive as the master drive by setting jumpers. Some drives have different jumper settings for a single master, as opposed to a master operating with a slave. Connect the drive to the connector at the end of the cable to reduce signal ringing that occurs when an unused connector is dangling at the end of the cable.
- If two drives are attached to the controller, jumper one drive as the master (or as a master operating with a slave), and jumper the second drive as a slave.

- If one drive is a hard disk and the second a CD-ROM drive, designate one drive as the slave drive by setting jumpers. It does not matter which drive is connected to which drive connection on the cable.
- If problems persist with two drives on a single controller, attach one drive at a time to verify that each drive works. Jumper the drive as master or single master, and use the drive connector at the end of the IDE ribbon cable to attach the drive. Verify that each drive works, then jumper the drives back to a master and slave configuration.
- If the drive is a disk drive, use the BIOS setup utility to ensure that the drive type (which indicates the number of cylinders, heads, and sectors) is configured correctly. Some BIOS software might have a feature that automatically detects the drive type.
- If the drive is a CD-ROM drive, use the BIOS setup screen to configure the drive type as a CD-ROM drive, provided the BIOS software offers this capability.
- For many systems, IDE CD-ROM drives are only recognized by MS-DOS if an MS-DOS CD-ROM driver has been installed. Try another drive.

`The IDE disk or CD-ROM drive on your system was not found by the Solaris Device Configuration Assistant` *x86 Platform Edition* `diskette during the pre-booting phase.` (*x86 based systems only*)

Solution:

- If disks are disabled in the BIOS, use the Solaris Device Configuration Assistant *x86 Platform Edition* diskette to boot from the hard disk. For information on accessing the Solaris Device Configuration Assistant, see “x86: Accessing the Solaris 9 Device Configuration Assistant and PXE” on page 32.
- If the system has no disks, it might be a diskless client.

`The system hangs before displaying the system prompt.` (*x86 based systems only*)

Solution: You have hardware that is not supported. Check your hardware manufacturer’s documentation.

Booting From the Network, Error Messages

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

Description: This error occurs when you have two or more servers on a network responding to an install client’s boot request. The install client connects to the wrong boot server, and the installation hangs. The following specific reasons might cause this error to occur:

Cause: *Reason 1:*`/etc/bootparams` files might exist on different servers with an entry for this install client.

Solution: *Reason 1:* Ensure that servers on the network do not have multiple `/etc/bootparams` entries for the install client. If they do have multiple entries, remove duplicate client entries in the `/etc/bootparams` file on all install servers and boot servers except the one you want the install client to use.

Cause: *Reason 2:* Multiple `/tftpboot` or `/rplboot` directory entries might exist for this install client.

Solution: *Reason 2:* Ensure that servers on the network do not have multiple `/tftpboot` or `/rplboot` directory entries for the install client. If they do have multiple entries, remove duplicate client entries from the `/tftpboot` or `/rplboot` directories on all install servers and boot servers except the one you want the install client to use.

Cause: *Reason 3:* An install client entry might exist in the `/etc/bootparams` file on a server and an entry in another `/etc/bootparams` file that enables all systems to access the profile server. Such an entry resembles the following:

```
* install_config=profile_server:path
```

A line that resembles the previous entry in the NIS or NIS+ `bootparams` table can also cause this error.

Solution: *Reason 3:* If a wildcard entry is in the name service `bootparams` map or table (for example, `* install_config=`), delete it and add it to the `/etc/bootparams` file on the boot server.

`No network boot server. Unable to install the system. See installation instructions.` (*SPARC based systems only*)

Cause: This error occurs on a system that you are attempting to install from the network. The system is not set up correctly.

Solution: Ensure that you correctly set up the system to install from the network. See "Adding Systems to Be Installed From the Network With a CD Image" on page 167.

`prom_panic: Could not mount file system` (*SPARC based systems only*)

Cause: This error occurs when you are installing Solaris 9 from a network, but the boot software cannot locate the following:

- Solaris DVD, either the DVD or a copy of the DVD image on the install server
- Solaris Software 1 of 2 CD image, either the Solaris Software 1 of 2 CD or a copy of the Solaris Software 1 of 2 CD image on the install server

Solution: Ensure that the installation software is mounted and shared.

- If you are installing Solaris 9 from the install server's DVD-ROM or CD-ROM drive, ensure that the Solaris DVD or Solaris Software 1 of 2 is inserted in the CD-ROM drive, is mounted, and is shared in the `/etc/dfs/dfstab` file.

- If installing from a copy of the Solaris DVD image or Solaris Software 1 of 2 CD image on the install server's disk, ensure that the directory path to the copy is shared in the `/etc/dfs/dfstab` file.

See the man page, `install_server`.

`Timeout waiting for ARP/RARP packet...`(*SPARC based systems only*)

Cause: *Reason 1:* The client is trying to boot from the network, but it cannot find a system that knows about the client.

Solution: *Reason 1:* Verify the system's host name is in the NIS or NIS+ name service. Also, verify the `bootparams` search order in the boot server's `/etc/nsswitch.conf` file.

For example, the following line in the `/etc/nsswitch.conf` file indicates that JumpStart or the Solaris `suninstall` program first looks in the NIS maps for `bootparams` information. If the program does not find any information, the JumpStart program or the Solaris `suninstall` program looks in the boot server's `/etc/bootparams` file.

```
bootparams: nis files
```

Cause: *Reason 2:* The client's Ethernet address is not correct.

Solution: *Reason 2:* Verify that the client's Ethernet address in the install server's `/etc/ethers` file is correct.

Cause: *Reason 3:* In a custom JumpStart installation, the `add_install_client` command specifies the platform group that uses a specified server as an install server. If the wrong architecture value is used when using the `add_install_client`, this problem occurs. For example, the machine you want to install is a sun4u, but you used sun4m instead.

Solution: *Reason 3:* Rerun `add_install_client` with the correct architecture value.

`ip: joining multicasts failed on tr0 - will use link layer broadcasts for multicast` (*x86 based systems only*)

Cause: This error message is displayed when you boot a system with a token ring card. Ethernet multicast and token ring multicast do not work the same way. The driver returns this error message because an invalid multicast address was provided to it.

Solution: Ignore this error message. If multicast does not work, IP uses layer broadcasts instead and does not cause the installation to fail.

`Requesting Internet address for` *Ethernet_Address* (*x86 based systems only*)

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

Solution: Verify the system's host name is listed in the name service. If the system's host name is listed in the NIS or NIS+ name service, and the system continues to print this error message, try rebooting.

`RPC: Timed out No bootparams (whoami) server responding; still trying...` (*x86 based systems only*)

Cause: The client is trying to boot from the network, but it cannot find a system with an entry in the `/etc/bootparams` file on the install server.

Solution: Use `add_install_client` on the install server. Using this command adds the proper entry in the `/etc/bootparams` file, enabling the client to boot from the network.

`Still trying to find a RPL server...` (*x86 based systems only*)

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

Solution: On the install server, execute `add_install_client` for the system to be installed. The `add_install_client` command sets up an `/rplboot` directory, which contains the necessary network boot program.

`CLIENT MAC ADDR: FF FF FF FF FF FF` (*network installations with DHCP only*)

Cause: The DHCP server is not configured correctly. This error might occur if the options or macros are not correctly defined in the DHCP Manager software.

Solution: In the DHCP Manager software, verify that the options and macros are correctly defined. Confirm that the Router option is defined, and that the value of the Router option is correct for the subnet you are using for the network installation.

Booting From the Network, General Problems

`The system boots from the network, but from a system other than the specified install server.`

Cause: An `/etc/bootparams` and perhaps `/etc/ethers` entry exist on another system for the client.

Solution: On the name server, update the `/etc/bootparams` entry for the system that is being installed. The entry should conform to the following syntax:

install_system `root=`*boot_server*`:`*path* `install=`*install_server*`:`*path*

Also, ensure that only one `bootparams` entry is on the subnet for the install client.

`After you set up an install server and configure the system to install Solaris 9 from the network, the system still does not boot.` (*SPARC based systems only*)

Cause: The `tftpd` might not be running on the install server.

Solution: Be sure the `tftpd` daemon is running on the install server. Type the following command:

```
# ps -ef | grep tftpd
```

If this command does not return a line that indicates that 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.

`After setting up an install server and configuring the system to install from the network, the system still does not boot.` (*x86 based systems only*)

Cause: The `rpld` daemon might not be running on the install server.

Solution: Be sure the `rpld` daemon is running on the install server. Type the following command:

```
# ps -ef | grep rpld
```

If this command does not return a line indicating the `rpld` daemon is running, execute the following command:

```
# /usr/sbin/rpld
```

After making this change, try booting the system again.

Initial Installation of the Solaris 9 Operating Environment

`Initial installation fails`

Solution: If the Solaris installation fails, you must restart the installation. To restart the installation, boot the system from the Solaris DVD, Solaris Installation CD, the Solaris Software 1 of 2, or from the network.

You cannot uninstall the Solaris software after the software has been partially installed. You must restore your system from a backup or begin the Solaris installation process again.

`/cdrom/Solaris_9/SUNWxxxx/reloc.cpio: Broken pipe`

Description: This error message does not affect the installation.

Solution: Ignore the message and continue with the installation.

`WARNING: CHANGE DEFAULT BOOT DEVICE` *(x86 based systems only)*

Cause: This is an informational message. The default boot device set in the system's BIOS might be set to a device that requires you to use the Solaris Device Configuration Assistant *x86 Platform Edition* diskette to boot the system.

Solution: Continue with the installation and, if necessary, change the system's default boot device specified in the BIOS after you install the Solaris software to a device that does not require the Solaris Device Configuration Assistant *x86 Platform Edition* diskette.

▼ x86: To Check IDE Disk for Bad Blocks

IDE disk drives do not automatically map out bad blocks like other drives supported by Solaris software. Before installing Solaris 9 on an IDE disk, you might want to perform a surface analysis on the disk. To perform surface analysis on an IDE disk, follow this procedure.

1. **Boot to the installation media in single-user mode.**

```
 # b -s
```

2. **Start the format program.**

```
# format
```

3. **Specify the IDE disk drive on which you want to perform a surface analysis.**

```
# cxdy
```

c*x* Is the controller number

d*y* Is the device number

4. **You need an `fdisk` partition.**

   - If a Solaris `fdisk` partition already exists, proceed to Step 5.
   - If a Solaris `fdisk` partition does not exist, use the `fdisk` command to create a Solaris partition on the disk.

```
format> fdisk
```

5. **Type:**

```
format> analyze
```

6. **Type:**

```
analyze> config
```

The current settings for a surface analysis are displayed.

a. **If you want to change settings, type:**

   ```
   analyze> setup
   ```

7. **Type:**

   ```
   analyze> type_of_surface_analysis
   ```

   type_of_surface_analysis   Is read, write, or compare

   If `format` finds bad blocks, it re-maps them.

8. **Type:**

   ```
   analyze> quit
   ```

9. **Do you want to specify blocks to re-map?**

   - If no, go to Step 10.
   - If yes, type:

     ```
     format> repair
     ```

10. **Type:**

    ```
    quit
    ```

    The `format` program quits.

11. **To restart the media in multiuser mode, type:**

    ```
    ok b
    ```

Upgrading the Solaris 9 Operating Environment

Upgrading, Error Messages

`No upgradable disks`

Cause: A swap entry in the `/etc/vfstab` file is causing the upgrade to fail.

Solution: Comment out the following lines in the `/etc/vfstab` file:

- All swap files and slices on disks not being upgraded
- Swap files that are no longer present
- Any unused swap slices

`usr/bin/bzczt not found`

Cause: Solaris Live Upgrade fails because of needing a patch cluster.

Solution: A patch is needed to install Solaris Live Upgrade. Go to http://sunsolve.sun.com for the patch.

`Upgradeable Solaris root devices were found, however, no suitable partitions to hold the Solaris install software were found. Upgrading using the Solaris Installer is not possible. It might be possible to upgrade using the Solaris Operating Environment 1 of 2 CDROM. (x86 based systems only)`

Cause: You cannot upgrade with Solaris *x86 Platform Edition* Installation CD because you do not have enough space.

Solution: To upgrade, you can either create a swap slice that is larger than or equal to 512 Mbytes or use another method of upgrading such as the following:

- The Solaris Web Start program from Solaris DVD or a net installation image
- The Solaris `suninstall` program from the Solaris Software 1 of 2 *x86 Platform Edition* CD
- Custom JumpStart

Upgrading, General Problems

`The upgrade option is not presented even though there is a version of Solaris software that's upgradable on the system.`

Cause: *Reason 1:* The `/var/sadm` directory is a symlink or it is mounted from another file system.

Solution: *Reason 1:* Move the `/var/sadm` directory into the root (`/`) or `/var` file system.

Cause: *Reason 2:* The `/var/sadm/softinfo/INST_RELEASE` file is missing.

Solution: *Reason 2:* Create a new `INST_RELEASE` file by using the following template:

```
OS=Solaris
VERSION=x
REV=0
```

x Is the version of Solaris software on the system

Cause: *Reason 3:* SUNWusr is missing from `/var/sadm/softinfo`.

Solution: *Solution 3:* You need to do an initial installation. The Solaris software is not upgradable.

`Couldn’t shut down or initialize the md driver`

Solution:

- If not a mirror, comment out in the `vsftab` file.
- If a mirror, break the mirror and reinstall.

`The upgrade fails because the Solaris installation program cannot mount a file system.`

Cause: During an upgrade, the script attempts to mount all the file systems that are listed in the system’s `/etc/vfstab` file on the root (`/`) file system that is being upgraded. If the installation script cannot mount a file system, it fails and exits.

Ensure that all file systems in the system’s `/etc/vfstab` file can be mounted. Comment out any file systems in the `/etc/vfstab` file that cannot be mounted or that might cause the problem so that the Solaris `suninstall` program does not try to mount them during the upgrade. Any system-based file systems that contain software to be upgraded (for example, `/usr`) cannot be commented out.

`The upgrade fails`

Description: The system does not have enough space for the upgrade.

Cause: Check Chapter 5 for the space problem and see if you can fix it without using auto-layout to reallocate space.

`Problems upgrading mirrored roots`

Solution: If you have problems upgrading when using Solaris Volume Manager mirrored roots, see “Troubleshooting Solaris Volume Manager (Tasks)” in *Solaris Volume Manager Administration Guide*.

▼ To Continue Upgrading After a Failed Upgrade

The upgrade fails and the system cannot be soft-booted. The failure is for reasons beyond your control, such as a power failure or a network connection failure.

1. **Reboot the system from the Solaris DVD, Solaris Installation CD, the Solaris Software 1 of 2 CD, or from the network.**

2. **Choose the upgrade option for installation.**

   The Solaris installation program determines if the system has been partially upgraded and continues the upgrade.

System Panics When Upgrading With Solaris Live Upgrade Running Veritas VxVm

When using Solaris Live Upgrade while upgrading and running Veritas VxVM, the system panics on reboot unless you upgrade by using the following procedure. The problem occurs if packages do not conform to Solaris advanced packaging guidelines.

1. **Create an inactive boot environment. See "Creating a New Boot Environment" on page 414.**

2. **Before upgrading the inactive boot environment, you must disable the existing Veritas software on the inactive boot environment.**

   a. **Mount the inactive boot environment.**

   ```
   # lumount inactive_boot_environment_name  mount_point
   ```

   For example:

   ```
   # lumount solaris8 /.alt.12345
   ```

   b. **Change to the directory that contains the `vfstab`, for example:**

   ```
   # cd /.alt.12345/etc
   ```

   c. **Make a copy of the inactive boot environment's `vfstab` file, for example:**

   ```
   # cp vfstab vfstab.501
   ```

   d. **In the copied `vfstab`, comment out all Veritas file system entries, for example:**

   ```
   #  sed  '/vx\/dsk/s/^/#/g'  < vfstab >  vfstab.novxfs
   ```

   The first character of each line is changed to #, which makes the line a comment line. Note that this comment line is different than the system file comment lines.

   e. **Copy the changed `vfstab` file, for example:**

   ```
   # cp vfstab.novxfs vfstab
   ```

   f. **Change directories to the inactive boot environment's system file, for example:**

   ```
   # cd /.alt.12345/etc
   ```

   g. **Make a copy of the inactive boot environment's system file, for example:**

   ```
   # cp system system.501
   ```

   h. **Comment out all "forceload:" entries that include `drv/vx`.**

   ```
   # sed '/forceload:   drv\/vx/s/^/*/' <system> system.novxfs
   ```

The first character of each line is changed to *, which makes the line a command line. Note that this comment line is different than the `vfstab` file comment lines.

i. **Change directories to the `install-db` file on the inactive boot environment, for example:**

```
# cd /.alt.12345/etc
```

j. **Create the Veritas `install-db` file, for example:**

```
# touch vx/reconfig.d/state.d/install-db
```

k. **Unmount the inactive boot environment.**

```
# luumount inactive_boot_environment_name  mount_point
```

3. **Upgrade the inactive boot environment. See Chapter 36.**

4. **Activate the inactive boot environment. See "Activating a Boot Environment" on page 461.**

5. **Shut down the system.**

```
# init 0
```

6. **Boot the inactive boot environment in single-user mode:**

```
OK boot -s
```

Several messages and error messages that contain "vxvm" or "VXVM" are displayed that can be ignored. The inactive boot environment becomes active.

7. **Upgrade Veritas.**

a. **Remove the Veritas VRTSvmsa package from system, for example:**

```
# pkgrm VRTSvmsa
```

b. **Change directories to the Veritas packages.**

```
# cd /location_of_Veritas_software
```

c. **Add the latest Veritas packages to the system:**

```
#pkgadd -d 'pwd' VRTSvxvm VRTSvmsa VRTSvmdoc VRTSvmman VRTSvmdev
```

8. **Restore the original `vfstab` and system files:**

```
# cp /etc/vfstab.original /etc/vfstab
    # cp /etc/system.original /etc/system
```

9. **Reboot the system.**

```
# init 6
```

x86: Service Partition Not Created by Default on Systems With No Existing Service Partition

If you install the Solaris 9 operating environment on a system that does not currently include a Service partition, the installation program might not create a Service partition by default. If you want to include a Service partition on the same disk as the Solaris partition, you must recreate the Service partition before you install the Solaris 9 operating environment.

If you installed the Solaris 8 2/02 operating environment on a Sun LX50 system, the installation program might not have preserved the Service partition. If you did not manually edit the `fdisk` boot partition layout to preserve the Service partition, the installation program deleted the Service partition during the installation.

Note – If you did not specifically preserve the Service partition when you installed the Solaris 8 2/02 operating environment, you might not be able to recreate the Service partition and upgrade to the Solaris 9 operating environment.

If you want to include a Service partition on the disk that contains the Solaris partition, choose one of the following workarounds.

- To use the Solaris Web Start installation program to install from the Solaris 9 Installation CD, follow these steps.

  1. Delete the contents of the disk.
  2. Before you install, create the Service partition by using the Sun LX50 Diagnostics CD.

     For information on how to create the Service partition, see the *Sun LX50 Server User's Manual* and the Sun LX50 Knowledge Base at `http://cobalt-knowledge.sun.com`.
  3. Insert the Solaris 9 Installation CD in the CD-ROM drive.
  4. Begin installing the Solaris 9 operating environment.

     When the installation program detects the Service partition, the following message is displayed.

     ```
     The default layout for the bootdisk is one x86 Boot partition
     and a Solaris partition on the remaining space. The Service fdisk
     partition, if one exists, is also preserved by default.

     Select one of the following to continue:

     1) Use the default layout
     2) Run fdisk to manually edit the disk
     3) Exit
     ```

```
Please make a selection: [?]
```

5. Type 1 to use the default layout.

   The installation program preserves the Service partition and creates the x86 boot partition and the Solaris partition.

   Note – The Solaris Web Start installation program creates the x86 boot partition by removing 10 Mbytes from the Solaris `fdisk` partition. This utility prevents any existing `fdisk` partitions from being altered. Do not create this partition manually.

   ---

6. Complete the installation.

- To install from a network installation image or from the Solaris 9 DVD over the network, follow these steps.

  1. Delete the contents of the disk.
  2. Before you install, create the Service partition by using the Sun LX50 Diagnostics CD.

     For information on how to create the Service partition, see the *Sun LX50 Server User's Manual* and the Sun LX50 Knowledge Base at `http://cobalt-knowledge.sun.com`.
  3. Boot the system from the network.

     The Customize `fdisk` Partitions screen is displayed.
  4. To load the default boot disk partition layout, click Default.

     The installation program preserves the Service partition and creates the x86 boot partition and the Solaris partition.

- To use the `suninstall` program to install from the Solaris 9 Software 1 of 2 CD or from a network installation image on a boot server, follow these steps.

  1. Delete the contents of the disk.
  2. Before you install, create the Service partition by using the Sun LX50 Diagnostics CD.

     For information on how to create the Service partition, see the *Sun LX50 Server User's Manual* and the Sun LX50 Knowledge Base at `http://cobalt-knowledge.sun.com`.
  3. Boot the system.

     The installation program prompts you to choose a method for creating the Solaris partition.
  4. Select the `Use rest of disk for Solaris partition` option.

     The installation program preserves the Service partition and creates the Solaris partition.

5. Complete the installation.

APPENDIX **F**

Installing or Upgrading Remotely (Tasks)

This appendix describes how to use the Solaris Web Start program to install or upgrade to the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive.

Note – If you are installing or upgrading the Solaris operating environment on a multi–domain server, refer to the system controller or system service processor documentation before beginning the installation process.

SPARC: Using the Solaris Web Start Program to Install or Upgrade From a Remote DVD-ROM or CD-ROM

If you want to install the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive, you can use a drive that is attached to another machine. Both machines must be connected to the same subnet. Use the following instructions to complete the installation.

▼ SPARC: To Install or Upgrade From a Remote DVD-ROM and CD-ROM

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

In the following procedure, the remote system with the DVD-ROM or CD-ROM is identified as *remote system*. The system that is the client to be installed is identified as *client system*.

1. **Identify a system that is running the Solaris operating environment and has a DVD-ROM or CD-ROM drive.**

2. **On the** *remote system* **with the DVD-ROM or CD-ROM drive, insert the Solaris DVD or the Solaris** *SPARC Platform Edition* **Installation CD in the drive.**

   The Volume Manager mounts the disc.

3. **On the remote system, change directories to the DVD or CD where the `add_install_client` command is located.**

   - For DVD media, type:

     ```
     remote system# cd /cdrom/cdrom0/s0/Solaris_9/Tools
     ```

   - For CD media, type:

     ```
     remote system# cd /cdrom/cdrom0/s0
     ```

4. **On the remote system, add the system that you want to install as a client.**

   - For DVD media, type:

     ```
     remote system# ./add_install_client  \
     client_system_name arch
     ```

   - For CD media, type:

     ```
     remote system# ./add_install_client -s remote_system_name: \
     /cdrom/cdrom0/s0 client_system_name arch
     ```

   remote_system_name The name of the system with the DVD-ROM or CD-ROM drive

   client_system_name The name of the machine you want to install

   arch The platform group of the machine you want to install, for example sun4u. On the system that you want to install, find the platform group by using the `uname -m` command.

5. **Boot the *client system* that you want to install.**

   ```
   client system: ok boot net
   ```

   The Solaris Web Start installation begins.

6. **Follow the instructions to type system configuration information if needed.**

   - If you are using DVD media, follow the instructions on the screen to complete the installation. You are finished.
   - If you are using CD media, the machine reboots and the Solaris installation program begins. After the Welcome panel, the Specify Media panel appears with Network File System selected. Proceed to Step 7.

7. **On the Specify Media panel, click Next.**

   The Specify Network File System Path panel appears and the text field contains the installation path.

   client_system_ip_address:/cdrom/cdrom0/s0

8. **On the remote system where the DVD or CD is mounted, change directories to `root`.**

   ```
   remote system# cd /
   ```

9. **On the remote system, check for the path to the slice that has been shared.**

   ```
   remote system# share
   ```

10. **On the remote system, unshare the Solaris DVD or Solaris Installation CD by using the path that is found in Step 9. If paths lead to two slices, `unshare` both slices.**

    ```
    remote system# unshare absolute_path
    ```

    absolute_path Is the absolute path shown in the `share` command

    In this example, slice 0 and slice 1 are unshared.

    ```
    remote system# unshare /cdrom/cdrom0/s0
    remote system# unshare /cdrom/cdrom0/s1
    ```

11. **Eject the Solaris Installation CD.**

    ```
    remote system# eject cdrom
    ```

12. **On the remote system, insert the Solaris Software 1 of 2 in the CD-ROM drive.**

13. **On the remote system, export the Solaris Software 1 of 2 CD.**

    ```
    remote system# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
    ```

14. **On the client system that you are installing, continue the Solaris installation by clicking Next.**

15. **If the Solaris Web Start program prompts you to insert the Solaris Software 2 of 2 CD, repeat Step 9 through Step 14 to unshare the Solaris Software 1 of 2 CD and to export and install the Solaris Software 2 of 2 CD.**

16. **If the Solaris Web Start program prompts you to insert the Solaris Languages CD, repeat Step 9 through Step 14 to unshare the Solaris Software 2 of 2 CD and to export and install the Solaris Languages CD.**

    When you export the Solaris Languages CD, an installer window appears on the machine where the CD-ROM is mounted. Ignore the installer window while you install the Solaris Languages CD. After you complete the installation of the Solaris Languages CD, close the installer window.

APPENDIX G

Additional SVR4 Packaging Requirements (Reference)

This appendix is for system administrators who need to use the custom JumpStart program or Solaris Live Upgrade to install or remove packages, especially third-party packages. Following these packaging requirements keeps a custom JumpStart installation non-interactive and avoids modifying the currently running system so you can upgrade with Solaris Live Upgrade.

Packaging Requirements Overview

The following documentation references provide background information on packaging requirements.

- For the custom JumpStart program and Solaris Live Upgrade to work properly, packages must comply with the SVR4 packaging requirements. The *Application Packaging Developer's Guide* provides more specific information on packaging requirements and definitions of terminology. See especially the chapter: "Advanced Package Creation Techniques" in *Application Packaging Developer's Guide*
- For basic information on adding and removing packages and the installation administration file, see "Managing Software (Overview)" in *System Administration Guide: Basic Administration*. Also see the specific man pages.
- For detailed information on the commands referenced in this appendix, see the man pages, `dircmp`(1), `fssnap`(1M), `ps`(1), or `truss`(1).

Table G–1 lists information that applies to either Solaris Live Upgrade or the custom JumpStart program in this document.

TABLE G–1 Requirements Information

Installation Method	Requirements Documented
Solaris Live Upgrade	■ See SVR4 packaging requirements in the *Application Packaging Developer's Guide*. ■ See "Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements" on page 654. ■ See "Solaris Live Upgrade Inactive Boot Environment Compliance" on page 656.
Custom JumpStart program	■ See SVR4 packaging requirements in the *Application Packaging Developer's Guide*. ■ See "Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements" on page 654. ■ See "Upgrade Compliance With the Custom JumpStart Program" on page 658.

Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements

An inactive boot environment is a copy of the operating environment, not the currently running system. A package to be used by Live Upgrade or the custom JumpStart program must follow these requirements:

- Enable a custom JumpStart installation or upgrade with no user interaction
- Make no modification of the current running system, which is necessary when using Solaris Live Upgrade

The following list explains the requirements for inactive boot environment compliance.

- For an installation of an operating system to be successful, packages must recognize and correctly respect inactive boot environment specifiers.

  Packages can include absolute paths in their `pkgmap` file (package map). If these files exist, they are written relative to the `-R` option of the `pkgadd` command. Packages that contain both absolute and relative (relocatable) paths can be installed to an alternative root (/) as well. `$PKG_INSTALL_ROOT` is prepended to both absolute and relocatable files so all paths are resolved properly when being installed by `pkgadd`.
- Packages being installed using the `pkgadd -R` option or being removed using the `pkgrm -R` option must not alter the currently running system.

Any procedure scripts that are included in the packages being installed with the `pkgadd` command `-R` option or being removed using the `pkgrm` command `-R` option must not alter the currently running system. Any installation scripts that you provide must reference any directory or file prefixed with the `$PKG_INSTALL_ROOT` variable. The package must write all directories and files with the `$PKG_INSTALL_ROOT` prefix. The package must not remove directories without a `$PKG_INSTALL_ROOT` prefix. Table G–2 provides examples of correct scripting syntax.

TABLE G–2 Examples of Installation Script Syntax

Script Type	Correct Syntax	Incorrect Syntax

Bourne shell “if” statement fragments	`if [ -f ${PKG_INSTALL_ROOT}\`
	`/etc/myproduct.conf ] ; then`
		`if [ -f /etc/myproduct.conf ] ; \`
		` then`

Removing a file	`/bin/rm -f ${PKG_INSTALL_ROOT}\`
	`/etc/myproduct.conf`
		`/bin/rm -f /etc/myproduct.conf`

Changing a file	`echo "test=no" > ${PKG_INSTALL_ROOT}\`
	`/etc/myproduct.conf`
		`echo "test=no" > \`
		`/etc/myproduct.conf`

Differences Between `$PKG_INSTALL_ROOT` and `$BASEDIR` Overview

`$PKG_INSTALL_ROOT` is the location of the root (`/`) file system of the machine to which you are adding the package. The location is set to the `-R` argument of the `pkgadd` command. For example, if the following command is invoked, then `$PKG_INSTALL_ROOT` becomes `/a` during the installation of the package.

```
# pkgadd -R /a SUNWvxvm
```

`$BASEDIR` points to the *relocatable* base directory into which relocatable package objects are installed. Only relocatable objects are installed here. Non-relocatable objects (those that have *absolute* paths in the `pkgmap` file) are always installed relative to the inactive boot environment, but not relative to the `$BASEDIR` in effect. If a package has no relocatable objects, then the package is said to be an absolute package (or non-relocatable), and `$BASEDIR` is undefined and not available to package procedure scripts.

For example, suppose a package’s `pkgmap` file has two entries:

```
1 f none sbin/ls 0555 root sys 3541 12322 1002918510
1 f none /sbin/ls2 0555 root sys 3541 12322 2342423332
```

The `pkginfo` file has a specification for `$BASEDIR`:

```
BASEDIR=/opt
```

If this package is installed with the following command, then `ls` is installed in `/a/opt/sbin/ls`, but `ls2` is installed as `/a/sbin/ls2`.

```
# pkgadd -R /a SUNWtest
```

Solaris Live Upgrade Inactive Boot Environment Compliance

When using Solaris Live Upgrade and creating a new boot environment, avoid problems by following these guidelines.

- Your package procedure scripts must be independent of the current active operating environment. Procedure scripts define actions that occur at particular points during package installation and removal. Four procedure scripts can be created with these predefined names: preinstall, postinstall, preremove, and postremove. Package procedure scripts must be independent of the currently active operating environment because an inactive boot environment could be switched by using Solaris Live Upgrade.
  - These scripts must not start or stop any processes or depend on the output of commands such as `ps` or `truss`, which are operating-system dependent and report information about the currently running system.
  - Procedure scripts are free to use other standard UNIX commands such as `expr`, `cp`, and `ls` and other commands that facilitate shell scripting. But, the inactive boot environment must not be modified except within the rules outlined in the section, "Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements" on page 654.
  - All scripts must be written in Bourne shell (`/bin/sh`). Bourne shell is the interpreter used by the `pkgadd` command to execute the procedure scripts.
  - Package procedure scripts must *not* invoke commands that were added in the 2.6 and later releases. For example, package procedure scripts cannot invoke the `pgrep` command. Since the 2.6 release, many commands have had additional features added. Package procedure scripts must not use any command options that did not exist in the 2.6 release. For example, the `-f` option of the `umount` command was added in the Solaris 7 release. To verify that a specific command or option is supported in the Solaris 2.6 release, see the *Solaris 2.6 Reference Manual AnswerBook* on `http://docs.sun.com`.
- All packages must pass `pkgchk` validation. After a package is created and before it is installed, it must be checked with the following command.

  ```
  # pkgchk -d dir_name pkg_name
  ```

dir_name Specifies the name of the directory where the package resides

pkg_name Specifies the name of the package

For example, if a package exists at /export/SUNWvxvm, then you would issue the following command.

```
# pkgchk -d /export SUNWvxvm
```

No errors should be displayed.

After a package is created, it must be tested by installing it in an inactive boot environment location using the -R *dir_name* option to pkgadd. After the package is installed, it must be checked for correctness using pkgchk, as in this example.

```
# pkgadd -d . -R /a SUNWvxvm
# pkgchk -R /a SUNWvxvm
```

No errors should be displayed.

- Also, packages must not execute commands delivered by the package itself. This is to maintain diskless compatibility and avoids running commands that might require shared libraries that are not installed yet.

These requirements for creating, modifying, and deleting files can be verified using a variety of commands. For example, the dircmp or fssnap commands can be used to verify that packages behave properly. Also, the ps command can be used for testing daemon compliance by making sure daemons are not stopped or started by the package. The truss, pkgadd -v, and pkgrm commands can test runtime package installation compliance, but might not work in all situations. In the following example, the truss command strips out all read-only, non-$TEMPDIR access and shows only non-read-only access to paths that do not lie within the specified inactive boot environment.

```
# TEMPDIR=/a; export TEMPDIR
# truss -t open /usr/sbin/pkgadd -R ${TEMPDIR} SUNWvxvm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v \
’open("’${TEMPDIR}
```

For detailed information on the commands referenced in this section, see the man pages dircmp(1), fssnap(1M), ps(1), truss(1), pkgadd(1M), pkgchk(1M), or pkgrm(1M).

Upgrade Compliance With the Custom JumpStart Program

Custom JumpStart program compliance ensures that packages can be added and removed while being part of the traditional Solaris installation utilities, which are the following:

- The custom JumpStart program
- Solaris suninstall program
- Solaris Web Start installation method

The custom JumpStart program compliance also ensures that the package can participate in Solaris upgrades. To be custom JumpStart program-compliant, a package also must follow inactive boot environment requirements that are outlined in "Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements" on page 654.

To be able to use the custom JumpStart program effectively, packages must be added or removed without the user being prompted for information. To prevent user interaction, set up a new administration file with the pkgadd command -a option. The -a option defines an installation administration file to be used in place of the default administration file. Using the default file might result in the user being prompted for more information. You can create an administration file that indicates to pkgadd that it should bypass these checks and install the package without user confirmation. For details, see the man pages, admin(4) or pkgadd(1M).

The following examples show how to use the pkgadd administration file.

- If no administration file is provided, pkgadd uses /var/sadm/install/admin/default. Using this file might result in user interaction.

  ```
  # pkgadd
  ```

- If a relative administration file is provided on the command line, pkgadd looks in /var/sadm/install/admin for the file name and uses it. In this example, the relative administration file is named nocheck and pkgadd looks for /var/sadm/install/admin/nocheck.

  ```
  # pkgadd -a nocheck
  ```

- If an absolute file is provided pkgadd uses it. In this example, pkgadd looks in /tmp for the nocheck administration file.

  ```
  # pkgadd -a /tmp/nocheck
  ```

The following is an example of an installation administration file that requires very little user interaction with the pkgadd utility. Unless the package requires more space than is available on the system, the pkgadd utility uses this file and installs the package without prompting the user for more information.

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
space=ask
setuid=nocheck
confiict=nocheck
action=nocheck
basedir=default
```

APPENDIX **H**

Upgrading to a Solaris Update Release (Tasks)

This chapter provides instructions on checking patches before upgrading to an update to the Solaris operating environment.

Upgrading to a Solaris Update Release

If you are already running the Solaris 9 operating environment and have installed individual patches, upgrading to a Solaris 9 Update release causes the following:

- Any patches that are supplied as part of the Solaris 9 Update release are reapplied to your system. You cannot back out these patches.
- Any patches that were previously installed on your system that are not included in the Solaris 9 Update release are removed.

The Patch Analyzer performs an analysis on your system to determine which patches, if any, will be removed by upgrading to the Solaris 9 Update release. The Patch Analyzer is available in the following formats.

- If you are using the Solaris Web Start program to upgrade, the Patch Analyzer dialog box appears. Select Yes to perform the analysis.
- If you are using the Solaris `suninstall` program to upgrade, select Analyze on the Patch Analysis dialog box to perform the analysis.
- If you are using a custom JumpStart installation or Solaris Live Upgrade to upgrade, run the `analyze_patches` script to perform the analysis. For detailed instructions, see "To Run the `analyze_patches` Script" on page 662.

After you perform the analysis, refer to "To Review the Patch Analyzer Output" on page 662 for detailed information about the patch analysis results.

▼ To Run the `analyze_patches` Script

Note – To run the `analyze_patches` script, the installed system and the Solaris DVD, Solaris Software CDs, or net image must be accessible by the script either through NFS or locally mounted media.

---

1. **Change to the `Misc` directory.**
   - SPARC: If the image is located on locally mounted media, type:

     ```
     # cd /cdrom/sol_9_Update_sparc/s0/Solaris_9/Misc
     ```

     In this command, *Update* is the actual Update identifier.
   - x86: If the image is located on locally mounted media, type:

     ```
     # cd /cdrom/sol_9_Update_x86/s2/Solaris_9/Misc
     ```

     In this command, *Update* is the actual Update identifier.
   - If the image is available on an NFS file system, type:

     ```
     # cd /NFS_mount_directory/Solaris_9/Misc
     ```

2. **Run the `analyze_patches` script.**

   ```
   # ./analyze_patches [-R rootdir] [-N netdir] [-D databasedir]
   ```

   `-R` *rootdir* — *rootdir* is the root of the installed system. The default is `/`.

   `-N` *netdir* — *netdir* is the path to the root of the OS image to be installed. The default is `/cdrom/cdrom0`. *netdir* is the path to the directory that contains the `Solaris_9` directory. You must use this option if you are running the `patch_analyzer` from an NFS mount point.

   `-D` *databasedir* — If the script is invoked from a directory other than the `Misc/` directory in the OS image, the program cannot find the database it uses for patch analysis. Use the `-D` option to supply the path to the database. Without this database, which is located in `Solaris_9/Misc/database` on the OS image, the script does not work properly.

▼ To Review the Patch Analyzer Output

After you perform the analysis, use these steps to review the output.

1. **Review the output of the Patch Analyzer.**

The Patch Analyzer provides a list of patches that will be removed, downgraded, accumulated, or obsoleted by other patches. Patch accumulations are similar to patch upgrades. The accumulated patch is removed and its fixes are delivered by a new patch. Messages such as the following are shown:

```
Patch 105644-03 will be removed.
Patch 105925 will be downgraded from -02 to -01.
Patch 105776-01 will be accumulated/obsoleted by patch 105181-05.
```

If the Patch Analyzer program does not provide a list, no action is taken against any patches that were previously installed on your system.

2. **Decide if the patch replacements and deletions are acceptable.**
   - If yes, upgrade the system.
   - If no, do not upgrade the system.

     At an update release, instead of upgrading, you can use the Solaris 9 Maintenance Update release to apply only patches to your system.

     Note – The Solaris 9 Maintenance Update is available on CD and by download. Instructions for applying patches are provided in the *Solaris 9 Maintenance Update Installation Guide*.

APPENDIX I

x86: Preparing to Boot From the Solaris Device Configuration Assistant or the Network (Tasks)

This appendix describes the following topics.

- "x86: Copying the Boot Software to a Diskette" on page 665
- "x86: Booting and Installing Over the Network With PXE" on page 667

x86: Copying the Boot Software to a Diskette

The Solaris Device Configuration Assistant is a program that enables you to perform various hardware configuration and booting tasks. The Solaris Device Configuration Assistant image is found in the Tools directory of either the Solaris *x86 Platform Edition* DVD or the Solaris Software 2 of 2 *x86 Platform Edition* CD. Use the following procedure to copy the boot image to a 3.5 diskette.

Note – You can boot directly from DVD or CD media or by using a net image with PXE. For information on these methods of booting, see "x86: Accessing the Solaris 9 Device Configuration Assistant and PXE" on page 32.

▼ x86: To Copy the Boot Software to a Diskette

Note – This procedure assumes that the system is running Volume Manager. If you are not using Volume Manager to manage diskettes and discs, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without Volume Manager.

---

1. **Log in as superuser on an x86 system to which a diskette drive is attached.**
2. **On the system with the DVD-ROM or CD-ROM drive, insert the Solaris** *x86 Platform Edition* **DVD or the Solaris Software 2 of 2** *x86 Platform Edition* **CD in the drive.**

   The Volume Manager mounts the disc.
3. **Change to the directory that contains the boot image.**
   - For DVD media, type:

     ```
     # cd /cdrom/sol_9_x86/s2/Solaris_9/Tools
     ```
   - For CD media, type:

     ```
     # cd /cdrom/sol_9_x86/Solaris_9/Tools
     ```
4. **Insert a blank diskette or a diskette that can be overwritten in the diskette drive.**
5. **Notify Volume Manager to check for new media.**

   ```
   # volcheck
   ```
6. **Format the diskette:**

   Caution – Formatting erases all data on the diskette.

   ```
   # fdformat -d -U
   ```
7. **Copy the file to the diskette.**

   ```
   # dd if=d1_image of=/vol/dev/aliases/floppy0 bs=36k
   ```
8. **Eject the diskette by typing `eject floppy` at the command line, and then manually ejecting the diskette from the drive.**

x86: Booting and Installing Over the Network With PXE

This section describes how to set up an x86-based system to install over the network without local boot media. This section describes the following topics.

- "What is PXE?" on page 667
- "x86: Guidelines for Booting With PXE" on page 667
- "Booting With PXE (Task Map)" on page 668
- "x86: Creating an x86 Install Server" on page 669
- "x86: Adding Systems to Install Over the Network By Using PXE" on page 672
- "x86: Booting the Client Over the Network By Using PXE" on page 676

What is PXE?

PXE network boot is a "direct" network boot. No boot media is required on the client system. With PXE, you can install an x86-based client over the network by using DHCP.

PXE network boot is available only for devices that implement the Intel Preboot Execution Environment specification. To determine if your system supports PXE network boot, see your hardware manufacturer's documentation.

The Solaris boot diskette is still available for systems that do not support PXE. The boot diskette image is available on the Solaris Software 2 of 2 *x86 Platform Edition* CD.

x86: Guidelines for Booting With PXE

To boot over the network by using PXE, you need the following systems.

- An install server
- A DHCP server
- An x86 client that supports PXE

When you are preparing to use PXE to install a client over the network, consider the following issues.

- Set up only one DHCP server on the subnet that includes the client system that you want to install. The PXE network boot does not work properly over subnets that include multiple DHCP servers.
- Some early versions of PXE firmware cannot boot the Solaris system. A system with these older versions can read the PXE network bootstrap program from a boot server, but the bootstrap does not transmit packets. To avoid this problem, upgrade the PXE firmware on the adapter. Obtain firmware upgrade information from the

adapter manufacturer's web site. Refer to the `elxl`(7D) and `iprb`(7D) man pages for more information.

Booting With PXE (Task Map)

Perform the following tasks to boot and install your system over the network by using PXE.

TABLE I–1 x86: Task Map: Booting From the Network By Using PXE

Task	Description	Instructions
Verify that your system supports PXE.	Confirm that your machine can use PXE to boot without local boot media.	Check your hardware manufacturer's documentation.
Choose an installation method.	The Solaris operating environment provides several methods for installation or upgrade. Choose the installation method that is most appropriate for your environment.	Chapter 3
Gather information about your system.	Use the checklist and complete the worksheet to collect all of the information that you need to install or upgrade.	Chapter 6
(Optional) Preconfigure system information.	You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.	Chapter 7
Create an install server.	Set up a Solaris server to install the Solaris operating environment from the network.	"x86: Creating an x86 Install Server" on page 669
Add systems to be installed over the network.	Use `add_install_client -d` to add DHCP support to install a class of client (of a certain machine type, for example) or a particular client ID.	"x86: Adding Systems to Install Over the Network By Using PXE" on page 672

TABLE I–1 x86: Task Map: Booting From the Network By Using PXE *(Continued)*

Task	Description	Instructions
Set up a DHCP server.	Plan for and configure your DHCP service.	"Planning for DHCP Service (Task)" in *System Administration Guide: IP Services*.
Create DHCP options for installation parameters and macros that include the options.	Use DHCP Manager or `dhtadm` to create the vendor options and macros that are output from the `add_install_client -d` command.	"Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86
Boot the client.	Instruct the client BIOS to boot from the network.	"x86: Booting the Client Over the Network By Using PXE" on page 676

x86: Creating an x86 Install Server

The install server contains the installation image that is needed to install systems from the network. You must create an install server to install the Solaris software on a system from the network.

Note – If you want use the Solaris DVD media to set up an install server on a system that is running the Solaris 2.6 or 7 operating environment, you must first apply one of the following patches.

- Solaris 2.6 *Intel Platform Edition* operating environment - Patch ID 107619-03
- Solaris 7 *Intel Platform Edition* operating environment - Patch ID 107260-03

▼ To Create an x86 Install Server

Note – This procedure makes the following assumptions.

- You are creating an install server on an x86–based system. For instructions on how to use a SPARC system to serve Solaris (x86 Platform Edition) installation images, see "To Create a SPARC Install Server With SPARC or x86 DVD Media" on page 129.
- The system has a CD-ROM drive.
- The system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to *System Administration Guide: Devices and File Systems* for detailed information about managing removable media without the Volume Manager.

---

1. **On the system that is to become the install server, become superuser.**

   The system must include a CD-ROM drive and be part of the site's network and name service. If you use a name service, the system must already be in a name service, such as NIS, NIS+, DNS, or LDAP. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **Insert the Solaris Software 1 of 2 CD in the system's drive.**

3. **Create a directory for the CD image.**

   ```
   # mkdir -p install_dir_path
   ```

   install_dir_path   Specifies the directory where the CD image is to be copied

4. **Change to the `Tools` directory on the mounted disc.**

   ```
   # cd /cdrom/cdrom0/s2/Solaris_9/Tools
   ```

   In the previous example, **`cdrom0`** is the path to the drive that contains the Solaris operating environment CD media.

5. **Copy the image in the drive to the install server's hard disk.**

   ```
   # ./setup_install_server install_dir_path
   ```

   install_dir_path   Specifies the directory where the CD image is to be copied

   ---

   Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris Software disc images. To determine available disk space, use the `df -kl` command.

   ---

6. **Change directories to root (`/`).**

```
# cd /
```

7. **Eject the Solaris Software 1 of 2 CD.**

8. **Insert the Solaris Software 2 of 2 CD in the system's CD-ROM drive.**

9. **Change to the `Tools` directory on the mounted CD:**

   ```
   # cd /cdrom/cdrom0/Solaris_9/Tools
   ```

10. **Copy the CD in the CD-ROM drive to the install server's hard disk.**

    ```
    # ./add_to_install_server install_dir_path
    ```

    install_dir_path   Specifies the directory where the CD image is to be copied

11. **Change directories to root (/).**

    ```
    # cd /
    ```

12. **Eject the Solaris Software 2 of 2 CD.**

13. **Insert the Solaris Languages CD in the system's CD-ROM drive.**

14. **Change to the `Tools` directory on the mounted CD:**

    ```
    # cd /cdrom/cdrom0/Tools
    ```

15. **Copy the CD in the CD-ROM drive to the install server's hard disk.**

    ```
    # ./add_to_install_server install_dir_path
    ```

    install_dir_path   Specifies the directory where the CD image is to be copied

16. **Change directories to root (/).**

    ```
    # cd /
    ```

17. **If you want to patch the files that are located in the miniroot (/*install_dir_path*/`Solaris_9/Tools/Boot`) on the net install image, use the `patchadd -C` command to patch these files. You might need to patch a file if a boot image has problems.**

Caution – Don't use the `patchadd -C` command unless you have read the `Patch README` instructions or have contacted your local Sun support office.

EXAMPLE I–1 x86: Creating an x86 Install Server With x86 CD Media

The following example illustrates how to create an install server by copying the following CDs to the install server's `/export/home/cdx86` directory.

- Solaris Software 1 of 2 *x86 Platform Edition* CD

- Solaris Software 2 of 2 *x86 Platform Edition* CD
- Solaris *x86 Platform Edition* Languages CD

```
# mkdir -p /export/home/cdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/s2
# ./modify_install_server -p /export/home/cdx86 /cdrom/cdrom0/s2
```

In this example, each CD is inserted and automatically mounted before each of the commands. After each command, the CD is removed.

x86: Adding Systems to Install Over the Network By Using PXE

After you create an install server, you must set up each system that you want to install from the network. Each system that you want to install needs to find the following:

- An install server
- A DHCP server
- The `sysidcfg` file if you use a `sysidcfg` file to preconfigure system information
- A name server if you use a name service to preconfigure system information
- The profile in the JumpStart directory on the profile server if you are using the custom JumpStart installation method

Use the following `add_install_client` procedure for set up an x86 client to install from the network by using PXE.

▼ To Add a Client to Install Over the Network With DHCP

1. **On the install server, become superuser.**
2. **If you use the NIS, NIS+, DNS, or LDAP name service, verify that the following information about the system to be installed has been added to the name service:**
   - Host name

- IP address
- Ethernet address

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

3. **Change to the `Tools` directory on the Solaris 9 CD image on the install server:**

```
# cd /install_dir_path/Solaris_9/Tools
```

install_dir_path Specifies the path to the `Tools` directory

4. **Set up the client system to be installed from the network.**

```
# ./add_install_client -d [-s install_server:install_dir_path] \
[-c jumpstart_server:jumpstart_dir_path]    [-p sysid_server:path] \
  [-t boot_image_path] [-b "boot-property=value"] \
[-e ethernet_address] client_name platform_group
```

`-d` Specifies that the client is to use DHCP to obtain the network install parameters. If you use the `-d` only, the `add_install_client` command sets up the installation information for client systems of the same class, for example, all x86 client machines. To set up the installation information for a specific client, use the `-d` with the `-e` option.

For more information about class-specific installations by using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86.

`-s` *install_server:install_dir_path* Specifies the name and path to the install server.

- *install_server* is the host name of the install server
- *install_dir_path* is the absolute path to the Solaris 9 CD image

`-c` *jumpstart_server* : *jumpstart_dir_path* Specifies a JumpStart directory for custom JumpStart installations. *jumpstart_server* is the host name of the server on which the JumpStart directory is located. *jumpstart_dir_path* is the absolute path to the JumpStart directory.

`-p` *sysid_server* : *path* Specifies the path to the `sysidcfg` file for preconfiguring system information. *sysid_server* is either a valid host name or an IP address for the server that contains the file. *path* is the absolute path to the directory containing the `sysidcfg` file.

`-t` *boot_image_path*

Specifies the path to an alternate boot image if you want to use a boot image other than the image in the Tools directory on the Solaris 9 net installation image, CD, or DVD.

`-b` *"boot-property=value"*

x86 based systems only: Enables you to set a boot property variable that you want to use to boot the client from the network. The `-b` must be used with the `-e` option.

See the `eeprom`(1M) man page for descriptions of boot properties.

`-e` *ethernet_address*

Specifies the ethernet address of the client that you want to install. This option enables you to set up the installation information to use for a specific client.

For more information about client-specific installations by using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86.

client_name

Is the name of the system to be installed from the network. This name is *not* the host name of the install server.

platform_group

Is the platform group of the system to be installed. A detailed list of platform groups appears in Appendix C.

The previous command outputs the vendor options and macros that you need to add to you DHCP server. See "Creating DHCP Options and Macros for Solaris Installation Parameters" on page 86 for instructions about how to define these vendor options and macros in your DHCP server.

EXAMPLE I–2 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP and PXE (CDs)

The following example illustrates how to add an x86 install client to an install server when you are using DHCP to set installation parameters on the network. The `-d` option is used to specify that clients are to use the DHCP protocol for configuration.

EXAMPLE I–2 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP and PXE (CDs) *(Continued)*

The DHCP class name SUNW.i86pc indicates that this command applies to all Solaris x86 network boot clients, not just a single client. The `-s` option is used to specify that the clients are to be installed from the install server that is named `rosemary`. This server contains a Solaris Software 1 of 2 *x86 Platform Edition* CD image in `/export/home/cdx86`.

For more information on how to use DHCP to set installation parameters for network installations, see "Supporting Solaris Network Installation with the DHCP Service (Task Map)" in *System Administration Guide: IP Services*.

```
x86_install_server# cd /export/boot/Solaris_9/Tools
x86_install_server# ./add_install_client -d -s rosemary:/export/home/cdx86 \
SUNW.i86pc i86pc
```

EXAMPLE I–3 x86: Specifying a Serial Console to Use During a Network Installation (CDs)

The following example illustrates how to add an x86 install client to an install server and specify a serial console to use during the installation. This example sets up the install client in the following manner.

- The `-d` option indicates that the client is set up to use DHCP to set installation parameters.
- The `-e` option indicates that this installation occurs only on the client with the ethernet address 00:07:e9:04:4a:bf.
- The `-b` option instructs the installation program to use the serial port `ttya` as an input and an output device.

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" i86pc
```

For a complete description of the boot property variables that you can use with the `-b` option, see the `eeprom`(1M) man page.

EXAMPLE I–4 x86: Specifying a Boot Device to Use During a Network Installation (CDs)

The following example illustrates how to add an x86 install client to an install server and specify a boot device to use during the installation. If you specify the boot device when you set up the install client, you are not prompted for this information by the Device Configuration Assistant during the installation.

This example sets up the install client in the following manner.

- The `-d` option indicates that the client is set up to use DHCP to set installation parameters.

- The `-e` option indicates that this installation occurs only on the client with the ethernet address 00:07:e9:04:4a:bf.
- The first two uses of the `-b` option instruct the installation program to use the serial port `ttya` as an input and an output device.
- The third use of the `-b` option instructs the installation program to use a specific boot device during the installation.

Note – The value of the boot device path varies based on your hardware.

- The i86pc platform name indicates that the client is an x86-based system.

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

For a complete description of the boot property variables that you can use with the `-b` option, see the `eeprom`(1M) man page.

x86: Booting the Client Over the Network By Using PXE

To install the system over the network, you must instruct the client system to boot over the network. Enable PXE network boot on the client system by using the BIOS setup program in the system BIOS, the network adapter BIOS, or both. On some systems, you must also adjust the boot device priority list so that network boot is attempted before booting from other devices. See the manufacturer's documentation for each setup program, or watch for setup program instructions during boot.

During the installation, answer the system configuration questions, if you are prompted.

After the system boots and installs over the network, instruct the system to boot from the disk drive on subsequent boots.

Some PXE-capable network adapters have a feature that enables PXE boot if you type a particular keystroke in response to a brief boot-time prompt. This feature enables you to boot over the network with PXE without modifying the PXE settings. If your adapter does not have this feature, disable PXE in the BIOS setup when the system reboots after installation. The system then boots from the disk drive.

APPENDIX **J**

Solaris 9 Updates

This appendix describes changes to this book for Solaris 9 update releases.

Solaris 9 9/02 Release

Minor changes were made to fix a bug.

Solaris 9 12/02 Release

The following list describes new features and changes to this book for the Solaris 9 12/02 release.

- When using the Solaris Flash installation feature, several new procedures and examples have been added.
  - The `flarcreate` command is used to create a Solaris Flash archive. The command has been updated with new options that increase your flexibility to define archive contents when creating an archive. You now can exclude more than one file or directory. From an excluded directory, you can restore a subdirectory or file. This feature is useful when you want to exclude large data files that you do not want cloned. For information on how to use these options, see "Customizing an Archive's Files and Directories" on page 214.
  - New procedures for splitting, merging, or extracting information about the archive have been added. For information on these procedures, see "Administering Solaris Flash Archives" on page 238.

- The Solaris installation programs now support LDAP Version 2 profiles, which enable you to configure your system to use a proxy credential level. During the Solaris Web Start or Solaris `suninstall` program installation programs, you can specify the LDAP proxy bind distinguished name and proxy bind password. With any installation method, you can preconfigure LDAP before installation by using the `proxy_dn` and `proxy_password` keywords in the sysidcfg file. For information on preconfiguring these profiles, see "Preconfiguring With the `sysidcfg` File" on page 65.
- When using Solaris Live Upgrade, new procedures and examples have been added, which are the following:
  - Creating a profile
  - Testing the profile
  - Using the profile to upgrade or install a Solaris Flash archive

  For more information, see Chapter 36.
- This book now contains procedures and descriptions that relate to x86 based systems.
- Minor changes were made to fix bugs.

Solaris 9 4/03 Release

The following list describes new features and changes to this book for the Solaris 9 4/03 release.

- The Solaris Flash installation feature provides new enhancements for this Solaris release.
  - A Solaris Flash installation can now update a clone system with minor changes. If you have a clone system and want to update it with minor changes, you can create a differential archive that contains only the differences between two images, the original master image and an updated master image. When you update a clone system with a differential archive, only the files specified in the differential archive are changed. The installation is restricted to clone systems that contain software consistent with the original master image. You use the custom JumpStart installation method to install a differential archive on a clone system. For procedures on creating a differential archive, see "To Create a Solaris Flash Differential Archive With an Updated Master Image" on page 231.
  - Special scripts can now be run for configuration of the master or clone or run to validate the archive. These scripts enable you to do the following tasks.
    - Configure applications on clone systems. You can use a custom JumpStart script for some uncomplicated configurations. For more complicated configurations, special configuration file processing might be necessary on the master system or before or after installation on the clone system. Also,

        local preinstallation and postinstallation scripts can reside on the clone and protect local customizations from being overwritten by the Solaris Flash software.

    - Identify non-clonable, host-dependent data that enables you to make the flash archive host independent. Host independence is enabled by modifying such data or excluding it from the archive. An example of host-dependent data is a log file.
    - Validate software integrity in the archive during creation
    - Validate the installation on the clone system

    For information on creating scripts, see "Creating Customization Scripts" on page 221.

- In the Solaris 9 4/03 operating environment, the Solaris Web Start and `suninstall` installation programs use a new default boot-disk partition layout to accommodate the Service partition on x86–based systems. If your system currently includes a Service partition, the new default boot-disk partition layout enables you to preserve this partition.

  For more information, see "x86: Change in Default Boot-Disk Partition Layout" on page 48.

- Minor changes were made to fix bugs.

Solaris 9 8/03 Release

The following list describes new features and changes to this book for the Solaris 9 8/03 release.

- Solaris Live Upgrade provides new enhancements for this Solaris release.
    - Solaris Live Upgrade uses Solaris Volume Manager technology to create a duplicate boot environment that contains file systems with RAID-1 volumes (mirrors). The mirror provides data redundancy for any file systems including the root (/) file system. With the `lucreate` command, you can create mirrored file systems that contain up to three submirrors.

      For an overview, see "Creating a Boot Environment With Mirrored File Systems" on page 382. For procedures, see "To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 435.

    - When creating a boot environment with the `lucreate` command, you can customize the content of files and directories that are copied. You can exclude some files and directories that would normally be copied from the original boot environment. If you have excluded a directory, you can also include specified files and subdirectories under that directory. For planning information, see "Customizing a New Boot Environment's Content" on page 405. For procedures, see “To Create a Boot Environment and Customize the Content (Command-Line Interface)” on page 440.

  - A Solaris Flash differential archive can now be installed on a clone system by using Solaris Live Upgrade. To install a differential archive by using Solaris Live Upgrade, see “To Create a Profile to be Used by Solaris Live Upgrade” on page 451.

- You can now use the JumpStart installation method to create an empty boot environment when you install the Solaris operating environment. The empty boot environment can then populated with a Solaris Flash archive.
- Minor changes were made to fix bugs.

Solaris 9 12/03 Release

The following list describes new features and changes to this book for the Solaris 9 12/03 release.

- Instructions about how to configure a DHCP server are now included in this guide. You can configure a DHCP server to provide client configuration information for network installations. For more information about how to configure your DHCP server to support your network installations, see “Creating DHCP Options and Macros for Solaris Installation Parameters” on page 86.
- The Solaris operating environment enables you to install remote clients over a wide area network (WAN). The WAN boot installation method enables you to transmit an encrypted Solaris Flash archive over a public network to a remote SPARC based client. The WAN boot programs then install the client system by performing a custom JumpStart installation. To protect the integrity of the installation, you can use private keys to authenticate and encrypt data. You can also transmit your installation data and files over a secure HTTP connection by configuring your systems to use digital certificates.
  - For overview information about the WAN boot installation method, see Chapter 41.
  - For information about how to plan your WAN boot installation, see Chapter 42.
  - For instructions about how to prepare your network for a WAN boot installation, see Chapter 43.
  - For instructions about how to install a remote client over a WAN, see Chapter 44.
  - For examples of the tasks that are required for a WAN boot installation, see Chapter 45.
  - For reference information about the commands and files that are required for a WAN boot installation, see Chapter 46.

- Minor changes were made to fix bugs.

Solaris 9 4/04 Release

The following list describes new features and changes to this book for the Solaris 9 4/04 release.

- The custom JumpStart installation method enables you to create RAID-1 volumes (mirrors) during the installation of the Solaris operating system. By mirroring file systems, you can protect your system by duplicating data over two physical disks. If one of the mirrored disks fails, the system data is still accessible from the second mirrored disk.
  - For overview information on how mirroring works, see Chapter 10.
  - For requirements and guidelines for creating mirrored file systems, see Chapter 11.
  - For an example of a JumpStart profile that creates RAID–1 volumes, see Example 26–11.
  - For a description of how to use the `filesys` profile keyword to create a mirror, see“`filesys` Profile Keyword (Creating Mirrored File Systems)” on page 358.
  - For a description of how to use the `metadb` profile keyword to create state database replicas, see “`metadb` Profile Keyword (Creating State Database Replicas)” on page 364.
- The `add_install_client` command enables you to set boot properties for Solaris x86 clients during a PXE network boot. The `-b` option enables you to perform the following tasks with the `add_install_client` command.
  - You can specify an alternate console to use during your network installation.
  - You can specify the device to use as the network boot device during the installation.
  - You can instruct the client to perform a fully automated custom JumpStart installation.

  For a description of the `-b` option to the `add_install_client` command, refer to one of the following sections.

  - For network installations that use DVD media, see “Adding Systems to Be Installed From the Network With a DVD Image” on page 137.
  - For network installations that use CD media, see “Adding Systems to Be Installed From the Network With a CD Image” on page 167.
- For Solaris Live Upgrade several new procedures and examples have been added.

  - A new procedure instructs you on how to install Solaris Live Upgrade packages. See "To Install Solaris Live Upgrade With the `pkgadd` Command" on page 412.
  - A new procedure instructs you on how to use Solaris Live Upgrade to create a Solaris Flash differential archive. See "To Create a Solaris Flash Differential Archive By Using Solaris Live Upgrade" on page 233.
  - A new procedure and an example instruct you on how to create empty boot environments. For the step by step procedure, see "To Create an Empty Boot Environment for a Solaris Flash Archive (Command-Line Interface)" on page 432. For the example, see "Example of Creating an Empty Boot Environment and Installing a Solaris Flash Archive (Command-Line Interface)" on page 495.
  - A new example provides commands for creating boot environments with RAID-1 volumes. See "Example of Migrating From an Existing Volume to a Solaris Volume Manager RAID-1 Volume (Command-Line Interface)" on page 494.
- A new section that describes how to boot an x86–based client over the network by using DHCP and PXE has been added. For instructions and examples about how to perform a network installation with PXE, see "x86: Booting and Installing Over the Network With PXE" on page 667.
- Minor changes were made to fix bugs.

Solaris 9 9/04 Release

Minor changes were made to fix bugs.

Glossary

3DES (Triple DES) Triple-Data Encryption Standard. A symmetric-key encryption method which provides a key length of 168 bits.

AES Advanced Encryption Standard. A symmetric 128-bit block data encryption technique. The U.S. government adopted the Rijndael variant of the algorithm as its encryption standard in October 2000. AES replaces DES encryption as the government standard.

archive A file that contains a collection of files that were copied from a master system. The file also contains identification information about the archive, such as a name and the date that you created the archive. After you install an archive on a system, the system contains the exact configuration of the master system.

An archive could be a differential archive which is Solaris Flash archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files specified and is restricted to systems that contain software consistent with the unchanged master image.

arrow keys One of the four directional keys on the numeric keypad.

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. You can use begin scripts only with custom JumpStart installations.

boot To load the system software into memory and start it.

`bootlog-cgi` The CGI program that enables a web server to collect and store remote client booting and installation console messages during a WAN boot installation.

boot environment A collection of mandatory file systems (disk slices and mount points) that are critical to the operation of the Solaris operating environment. These disk slices might be on the same disk or distributed across multiple disks.

The active boot environment is the one that is currently booted. Exactly one active boot environment can be booted. An inactive boot environment is not currently booted, but can be in a state of waiting for activation on the next reboot.

boot server A server system that provides client systems on the same network subnet with the programs and information that they need to start. A boot server is required to install over the network if the install server is on a different subnet than the systems on which Solaris software is to be installed.

certificate authority (CA) A trusted third-party organization or company that issues digital certificates used to create digital signatures and public-private key pairs. The CA guarantees that the individual granted the unique certificate is who she or he claims to be.

`certstore` A file that contains a digital certificate for a specific client system. During an SSL negotiation, the client might be asked to provide the certificate file to the server. The server uses this file to verify the identity of the client.

CGI (Common Gateway Interface) An interface by which external programs communicate with the HTTP server. Programs that are written to use CGI are called CGI programs or CGI scripts. CGI programs handle forms or parse output the server does not normally handle or parse.

checksum The result of adding a group of data items that are used for checking the group. The data items can be either numerals or other character strings that are treated as numerals during the checksum calculation. The checksum value verifies that communication between two devices is successful.

client In the client-server model for communications, the client is a process that remotely accesses resources of a compute server, such as compute power and large memory capacity.

clone system A system that you installed by using a Solaris Flash archive. The clone system has the same installation configuration as the master system.

cluster A logical collection of packages (software modules). The Solaris 9 software is divided into *software groups*, which are each composed of clusters and *packages*.

command line A string of characters that begins with a command, often followed by arguments, including options, file names, and other expressions, and terminated by the end-of-line character.

concatenation A RAID-0 volume. If slices are concatenated, the data is written to the first available slice until that slice is full. When that slice is full, the data is written to the next slice, serially. A concatenation provides no data redundancy unless contained in a mirror. See also RAID-0 volume.

Core software group A software group that contains the minimum software that is required to boot and run the Solaris operating environment on a system. Core includes some networking software and the drivers that are required to run the Common Desktop Environment (CDE) desktop. Core does not include the CDE software.

critical file systems File systems that are required by the Solaris operating environment. When you use Solaris Live Upgrade, these file systems are separate mount points in the `vfstab` of the active and inactive boot environments. Examples are `root` (/), `/usr`, `/var`, or `/opt`. These file systems are always copied from the source to the inactive boot environment.

custom JumpStart A type of installation in which the Solaris 9 software is automatically installed on a system based on a user-defined profile. You can create customized profiles for different types of users and systems. A custom JumpStart installation is a JumpStart installation you create.

custom probes file A file, which must be located in the same JumpStart directory as the `rules` file, is a Bourne shell script that contains two types of functions: probe and comparison. Probe functions gather the information you want or do the actual work and set a corresponding `SI_` environment variable you define. Probe functions become probe keywords. Comparison functions call a corresponding probe function, compare the output of the probe function, and return 0 if the keyword matches or 1 if the keyword doesn't match. Comparison functions become rule keywords. See also *rules file*.

decryption The process of converting coded data to plain text.

See also encryption.

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

DES (Data Encryption Standard) A symmetric-key encryption method developed in 1975 and standardized by ANSI in 1981 as ANSI X.3.92. DES uses a 56-bit key.

Developer Solaris Software Group

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

DHCP

Dynamic Host Configuration Protocol is an application-layer protocol that enables individual computers, or clients, on a TCP/IP network to extract an IP address and other network configuration information from a designated and centrally maintained DHCP server or servers. This facility reduces the overhead of maintaining and administering a large IP network.

differential archive

A Solaris Flash archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files specified and is restricted to systems that contain software consistent with the unchanged master image.

digital certificate

A nontransferable, nonforgeable, digital file issued from a third party that both communicating parties already trust.

disc

An optical disc, as opposed to a magnetic disk, which recognizes the common spelling that is used in the compact disc (CD) market. For example, a CD-ROM or DVD-ROM is an optical disc.

disk

A round platter, or set of platters, of a magnetized medium that is organized into concentric tracks and sectors for storing data such as files. See also disc.

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–size disks.

diskless client

A client on a network that relies on a server for all of its disk storage.

document root directory

The root of a hierarchy on a web server machine that contains the files, images, and data you want to present to users that are accessing the web server.

domain

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

domain name

The name that is assigned to a group of systems on a local network that share administrative files. The domain name is required for the Network Information Service (NIS) database to work properly. A domain name consists of a sequence of component names that are 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.

encryption
The process of protecting information from unauthorized use by making the information unintelligible. Encryption is based on a code, called a key, which is used to decrypt the information.

See also decryption.

End User Solaris Software Group
A software group that contains the Core software group plus the recommended software for an end user, including the Common Desktop Environment (CDE) and DeskSet software.

Entire Solaris Software Group
A software group that contains the entire Solaris 9 release.

Entire Solaris Software Group Plus OEM Support
A software group that contains the entire Solaris 9 release, plus additional hardware support for OEMs. This software group is recommended when installing Solaris software on SPARC based servers.

EISA
Extended Industry Standard Architecture. A type of bus on x86 based systems. EISA bus standards are "smarter" than ISA bus systems. Attached devices can be automatically detected when they have been configured with the "EISA configurator" program that was supplied with the system. See also ISA.

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

`/etc/netboot` directory
The directory on a WAN boot server that contains the client configuration information and security data that are required for a WAN boot installation.

`/export`
A file system on an OS server that is shared with other systems on a network. For example, the `/export` file system can contain the root file system and swap space for diskless clients and the home directories for users on the network. Diskless clients rely on the `/export` file system on an OS server to boot and run.

fallback
A reversion to the environment that ran previously. Use fallback when you are activating an environment and the boot environment that is designated for booting fails or shows some undesirable behavior.

`fdisk` partition
A logical partition of a disk drive that is dedicated to a particular operating system on x86 based systems. To install the Solaris software, you must set up at least one Solaris 9 `fdisk` partition on an x86 based system. x86 based systems allow up to four different `fdisk` partitions on a disk. These partitions can be used to hold individual operating systems. Each operating system must be located on a unique `fdisk` partition. A system can only have one Solaris `fdisk` partition per disk.

file server
A server that provides the software and file storage for systems on a network.

file system In the SunOS™ operating system, a tree-structured network of files and directories that you can access.

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. You use finish scripts with custom JumpStart installations.

format To put data into a structure or divide a disk into sectors for receiving data.

function key One of the 10 or more keyboard keys that are labeled F1, F2, F3, and so on that are mapped to particular tasks.

hard link A directory entry that references a file on disk. More than one such directory entry can reference the same physical file.

hashing The process of changing a string of characters into a value or key that represents the original string.

hash A number that is produced by taking some input and generating a number that is significantly shorter than the input. The same output value is always generated for identical inputs. Hash functions can be used in table search algorithms, in error detection, and in tamper detection. When used for tamper detection, hash functions are chosen such that it is difficult to find two inputs that yield the same hash result. MD5 and SHA-1 are examples of one-way hash functions. For example, a message digest takes a variable length input such as a disk file and reduces it to a small value.

HMAC Keyed hashing method for message authentication. HMAC is used with an iterative cryptographic hash function, such as MD5 or SHA-1, in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

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 particular domain (usually, this means within any single organization). A host name can be any combination of letters, numbers, and minus signs (`-`), but it cannot begin or end with a minus sign.

HTTP (Hypertext Transfer Protocol) (n.) The Internet protocol that fetches hypertext objects from remote hosts. This protocol is based on TCP/IP.

HTTPS A secure version of HTTP, implemented by using the Secure Sockets Layer, SSL.

initial installation An installation that overwrites the currently running software or initializes a blank disk.

An initial installation of the Solaris operating environment overwrites the system's disk or disks with the new version of the Solaris operating environment. If your system is not running the Solaris operating environment, you must perform an initial installation.

initial installation option An option that is presented by the Solaris Web Start program and the Solaris `suninstall` program that overwrites the disk(s) with a new version of Solaris. The initial installation option is presented for systems that can be upgraded. However, the disk(s) that contain the old version of Solaris software (including the local modifications) are overwritten if you choose the initial installation option.

install server A server that provides the Solaris 9 DVD or CD images from which other systems on a network can install Solaris (also known as a *media server*). You can create an install server by copying the Solaris 9 DVD or CD images to the server's hard disk.

IP address Internet protocol address. In TCP/IP, a unique 32-bit number that identifies each host in a network. An IP address consists of four numbers that are separated by periods (192.9.9.1, for example). 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 local system on the network (similar to a phone number). The numbers in a Class A IP address, for example, represent "network.local.local.local" and the numbers in a Class C IP address represent "network.network.network.local."

Class	Range (*xxx* is a number 0 to 255)	Number of Available IP Addresses
Class A	1.*xxx.xxx.xxx* - 126.*xxx.xxx.xxx*	Over 16 million
Class B	128.0.*xxx.xxx* - 191.255.*xxx.xxx*	Over 65,000
Class C	192.0.0.*xxx* - 223.255.255.*xxx*	256

IPv6 IPv6 is a version (version 6) of Internet Protocol (IP) that is designed to be an evolutionary step from the current version, IPv4 (version 4). Deploying IPv6, by using defined transition mechanisms, does not disrupt current operations. In addition, IPv6 provides a platform for new Internet functionality.

IPv6 is described in more detail in "IPv6 (Overview)" in *System Administration Guide: IP Services*.

ISA Industry Standard Architecture. A type of bus found in x86 based systems. ISA bus systems are “dumb” and provide no mechanism the system can use to detect and configure devices automatically. See also EISA.

job A user-defined task to be completed by a computer system.

JumpStart directory When you use a profile diskette for custom JumpStart installations, the JumpStart directory is the root directory on the diskette that contains all the essential custom JumpStart files. When you use a profile server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files.

JumpStart installation A type of installation in which the Solaris 9 software is automatically installed on a system by using the factory-installed JumpStart software.

Kerberos A network authentication protocol that uses strong, secret-key cryptography to enable a client and server to identify themselves to each other over an insecure network connection.

key The code for encrypting or decrypting data.

See also encryption.

`keystore` A file that contains keys shared by a client and server. During a WAN boot installation, the client system uses the keys to verify the integrity of, or decrypt the data and files transmitted from, the server.

LAN (local area network) A group of computer systems in close proximity that can communicate by way of some connecting hardware and software.

LDAP Lightweight Directory Access Protocol is a standard, extensible directory access protocol that is used by LDAP naming service clients and servers to communicate with each other.

locale A geographic or political region or community that shares the same language, customs, or cultural conventions (English for the U.S. is en_US, and English for the U.K. is en_UK).

logical device A group of physical slices on one or more disks that appear to the system as a single device. A logical device is called a volume in Solaris Volume Manager. A volume is functionally identical to a physical disk in the view of an application or file system.

manifest section A section of a Solaris Flash archive that is used to validate a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. This section is informational only. The sections lists the files in an internal format and cannot be used for scripting.

master system

A system that you use to create a Solaris Flash archive. The system configuration is saved in the archive.

MD5

An iterative cryptographic hash function that is used for message authentication, including digital signatures. The function was developed in 1991 by Rivest.

media server

See *install server*.

metadevice

See *volume*.

miniroot

The smallest possible bootable Solaris `root` file system. A miniroot contains a kernel and just enough software to install the Solaris environment on a hard disk. The miniroot is the file system that is copied to a machine in the initial installation.

mirror

See RAID-1 volume

mount

The process of accessing a directory from a disk that is attached to a machine that is making the mount request or a remote disk on a network. 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 workstation directory to which you mount a file system that exists on a remote machine.

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 that 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. Without a name service, each system has to maintain its own copy of the system information in the local `/etc` files. Sun supports the following name services: LDAP, NIS, and NIS+.

networked systems

A group of systems (called hosts) that are connected through hardware and software, so that 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.

network installation

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

NIS

The SunOS 4.0 (minimum) Network Information Service. A distributed network database that contains key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.

NIS+ The SunOS 5.0 (minimum) Network Information Service. NIS+ replaces NIS, the SunOS 4.0 (minimum) Network Information Service.

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

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

OS server A system that provides services to systems on a network. To serve diskless clients, an OS server must have disk space set aside for each diskless client's root file system and swap space (`/export/root`, `/export/swap`).

package A collection of software that is grouped into a single entity for modular installation. The Solaris 9 software is divided into *software groups*, which are each composed of *clusters* and packages.

panel A container for organizing the contents of a window, dialog box, or applet. The panel might collect and confirm user input. Panels might be used by wizards and follow an ordered sequence to fulfill a designated task.

patch analyzer A script that you can run manually or as part of the Solaris `suninstall` program or the Solaris Web Start program. The patch analyzer performs an analysis on your system to determine which (if any) patches will be removed by upgrading to a Solaris 9 Update.

platform group A vendor-defined grouping of hardware platforms for the purpose of distributing specific software. Examples of valid platform groups are i86pc and sun4u.

platform name The output of the `uname -i` command. For example, the platform name for the Ultra 60 is SUNW,Ultra-60.

Power Management Software that automatically saves the state of a system and turns it off after it is idle for 30 minutes. When you install the Solaris software on a system that complies with Version 2 of the U.S. Environmental Protection Agency's Energy Star guidelines—a sun4u SPARC system, for example—the Power Management software is installed by default. After a subsequent reboot, you are prompted to enable or disable the Power Management software.

Energy Star guidelines require that systems or monitors automatically enter a "sleep state" (consume 30 watts or less) after the system or monitor becomes inactive.

probe keyword A syntactical element that extracts attribute information about a system when using the custom JumpStart method to install. A probe keyword does not require you to set up a matching condition and run a profile as required for a rule. See also *rule*.

profile A text file that defines how to install the Solaris software when using the custom JumpStart method. For example, a profile defines which software group to install. Every rule specifies a profile that defines how a system is to be installed when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule. See also *rules file*.

profile diskette A diskette that contains all the essential custom JumpStart files in its root directory (JumpStart directory).

profile server A server that contains all the essential custom JumpStart files in a JumpStart directory.

private key The decryption key used in public-key encryption.

public key The encryption key used in public-key encryption.

public key cryptography A cryptographic system that uses two keys: a public key known to everyone, and a private key known only to the recipient of the message.

RAID-1 volume A class of volume that replicates data by maintaining multiple copies. A RAID-1 volume is composed of one or more RAID-0 volumes called submirrors. A RAID-1 volume is sometimes called a mirror.

RAID-0 volume A class of volume that can be a stripe or a concatenation. These components are also called submirrors. A strip or concatenation is the basic building blocks for mirrors.

/ (root) In a hierarchy of items, the one item from which all other items are descended. The root item has nothing above it in the hierarchy. / is the base directory from which all other directories stem, directly or indirectly. The root directory contains the directories and files critical for system operation, such as the kernel, device drivers, and the programs that are used to start (boot) a system.

rule A series of values that assigns one or more system attributes to a profile. A rule is used in a custom JumpStart installation.

`rules` file A text file that contains a rule for each group of systems or single systems that you want to install automatically. Each rule distinguishes a group of systems, based on one or more system attributes. The `rules` file links each group to a profile, which is a text file that defines how the Solaris 9 software is to be installed on each system in the group. A rules file is used in a custom JumpStart installation. See also *profile*.

`rules.ok` file A generated version of the `rules` file. The `rules.ok` file 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.

Secure Sockets Layer (SSL) A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

server A network device that manages resources and supplies services to a client.

SHA1 (Secure Hashing Algorithm) The algorithm operates on any input length less than 2^{64} to produce a message digest.

shareable file systems File systems that are user-defined files such as `/export/home` and `/swap`. These file systems are shared between the active and inactive boot environment when you use Solaris Live Upgrade. Shareable file systems contain the same mount point in the `vfstab` in both the active and inactive boot environments. Updating shared files in the active boot environment also updates data in the inactive boot environment. Shareable file systems are shared by default, but you can specify a destination slice, and then the file systems are copied.

slice The unit into which the disk space is divided by the software.

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 Solaris Software, Developer Solaris Software, or Entire Solaris Software, and for SPARC systems only, Entire Solaris Software Group Plus OEM Support.

Solaris 9 DVD or CD images The Solaris 9 software that is installed on a system, which you can access on the Solaris 9 DVDs or CDs or an install server's hard disk to which you have copied the Solaris 9 DVD or CD images.

Solaris Flash A Solaris installation feature that enables you to create an archive of the files on a system, known as the master system. You can then use the archive to install other systems, making the other systems identical in their configuration to the master system. See also *archive.*

Solaris `suninstall` program A command–line interface (CLI), menu-driven, interactive script that enables you to set up a system and install the Solaris 9 software on it.

Solaris Live Upgrade An upgrade method that enables a duplicate boot environment to be upgraded while the active boot environment is still running, thus eliminating downtime of the production environment.

Solaris Web Start program A graphical user interface (GUI) or command–line interface (CLI) installation program that uses wizard panels to guide you step-by-step through installing the Solaris 9 software and third–party software.

standalone A computer that does not require support from any other machine.

state database A state database stores information on disk about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a state database replica. The state database tracks the location and status of all known state database replicas.

state database replica A copy of a state database. The replica ensures that the data in the database is valid.

submirror See RAID-0 volume.

subnet A working scheme that divides a single logical network into smaller physical networks to simplify routing.

subnet mask A bit mask that is used to select bits from an Internet address for subnet addressing. The mask is 32 bits long and selects the network portion of the Internet address and 1 or more bits of the local portion.

swap space A slice or file that temporarily holds the contents of a memory area till it can be loaded back into memory. Also known as the `/swap` or `swap` file system.

`sysidcfg` file A file in which you specify a set of special system configuration keywords that preconfigure a system.

system configuration file (`system.conf`) A text file in which you specify the locations of the `sysidcfg` file and the custom JumpStart files you want to use in a WAN boot installation.

time zone Any of the 24 longitudinal divisions of the earth's surface for which a standard time is kept.

`truststore` A file that contains one or more digital certificates. During a WAN boot installation, the client system verifies the identity of the server that is trying to perform the installation by consulting the data in the `truststore` file.

unmount The process of removing access to a directory on a disk that is attached to a machine or to a remote disk on a network.

update An installation, or to perform an installation, on a system that changes software that is of the same type. Unlike an upgrade, an update might downgrade the system. Unlike an initial installation, software of the same type that is being installed must be present before an update can occur.

upgrade An installation that merges files with existing files and saves modifications where possible.

An upgrade of the Solaris operating environment merges the new version of the Solaris operating environment with the existing files on the system's disk or disks. An upgrade saves as many modifications as possible that you have made to the previous version of the Solaris operating environment.

upgrade option An option that is presented by the Solaris Web Start program and the Solaris `suninstall` program. The upgrade procedure merges the new version of Solaris with existing files on your disk(s). An upgrade also saves as many local modifications as possible since the last time Solaris was installed.

URL (Uniform Resource Locator) The addressing system used by the server and the client to request documents. A URL is often called a location. The format of a URL is *protocol://machine:port/document*.

A sample URL is `http://www.example.com/index.html`.

`/usr` A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large `/usr` file system with a server rather than maintaining a local copy minimizes the overall disk space that is required to install and run the Solaris 9 software on a system.

utility A standard program, usually furnished at no charge with the purchase of a computer, that does the computer's housekeeping.

`/var` A file system or directory (on standalone systems) that contains system files that are likely to change or grow over the life of the system. These files include system logs, `vi` files, mail files, and uucp files.

volume A group of physical slices or other volumes that appear to the system as a single logical device. A volume is functionally identical to a physical disk in the view of an application or file system.

In some command-line utilities, a volume is called a metadevice. Volume is also known as pseudo device or virtual device in standard UNIX terms.

Volume Manager A program that provides a mechanism to administer and obtain access to the data on DVD-ROMs, CD-ROMs, and diskettes.

`wanboot` program The second level boot program that loads the WAN boot miniroot, client configuration files, and installation files that are required to perform a WAN boot installation. For WAN boot installations, the `wanboot` binary performs tasks similar to the `ufsboot` or `inetboot` second level boot programs.

WAN (wide area network) A network that connects multiple local-area networks (LANs) or systems at different geographical sites by using telephone, fiber-optic, or satellite links.

WAN boot installation A type of installation that enables you to boot and install software over a wide area network (WAN) by using HTTP or HTTPS. The WAN boot installation method enables you to transmit an encrypted Solaris Flash archive over a public network and perform a custom JumpStart installation on a remote client.

WAN boot miniroot

A miniroot that has been modified to perform a WAN boot installation. The WAN boot miniroot contains a subset of the software in the Solaris miniroot.

See also miniroot.

WAN boot server

A web server that provides the configuration and security files that are used during a WAN boot installation.

`wanboot-cgi` program

The CGI program that retrieves and transmits the data and files that are used in a WAN boot installation.

`wanboot.conf` file

A text file in which you specify the configuration information and security settings that are required to perform a WAN boot installation.

Index