Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning

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

Part No: 817–5505–10
January 2005

041129@10536

Contents

Preface

This book describes how to install and upgrade the Solaris™ 10 Operating System (OS) on both networked and nonnetworked SPARC® and x86 architecture based systems.

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

Note – This Solaris release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC®, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`. This document cites any implementation differences between the platform types.

In this document the term "x86" refers to 64-bit and 32-bit systems manufactured using processors compatible with the AMD64 or Intel Xeon/Pentium product families. For supported systems, see the *Solaris 10 Hardware Compatibility List*.

Who Should Use This Book

This book is intended for system administrators responsible for installing the Solaris OS. 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 might need when you install the Solaris software.

TABLE P–1 Related Information

Information	Description
Solaris 10 Installation Guide: Basic Installations	Describes a basic OS installation with a graphical user interface (GUI).
Solaris 10 Installation Guide: Network-Based Installations	Describes how to perform a remote Solaris installation over a local area network or a wide area network.
Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations	Describes how to create the files and directories necessary to perform an unattended custom JumpStart™ installation. This book also describes how to create RAID-1 volumes during a JumpStart installation.
Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning	Provides planning information when using CD or DVD media to upgrade a system to the Solaris OS. This book also describes how to use Solaris Live Upgrade to create and upgrade new boot environments.
Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)	Provides instructions for creating Solaris Flash archives that are used to install the Solaris OS on multiple systems.
Chapter 23, "Backing Up and Restoring File Systems (Overview)," in *System Administration Guide: Devices and File Systems*	Describes how to back up system files and other system administration tasks.
Solaris 10 Release Notes	Describes any bugs, known problems, software that is being discontinued, and patches that are related to the Solaris release.
SPARC: *Solaris 10 Sun Hardware Platform Guide*	Contains information about supported hardware.
Solaris 10 Package List	Lists and describes the packages in the Solaris 10 OS.
x86: Solaris Hardware Compatibility List for x86 Platforms	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 used in this book.

TABLE P–2 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; 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	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

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	`#`

PART I Overall Planning of Your Solaris Installation or Upgrade

This part guides you through planning the installation or upgrade of the Solaris Operating System.

CHAPTER 1

Solaris Installation and Upgrade (Roadmap)

This chapter provides you with information about decisions you need to make before you install or upgrade the Solaris Operating System (OS). This chapter contains the following sections:

Note – This book uses the term *slice*, but some Solaris documentation and programs might refer to a slice as a partition.

x86: To avoid confusion, this book distinguishes between x86 `fdisk` partitions and the divisions within the Solaris `fdisk` partition. The x86 `fdisk` divisions are called partitions. The divisions within the Solaris `fdisk` partition are called slices.

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 OS when using any installation program. 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 1–1 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 20
Choose an installation program.	The Solaris OS provides several programs for installation or upgrade. Choose the installation method that is most appropriate for your environment.	"Choosing a Solaris Installation Method" on page 21

(Solaris installation program) Choose a default or custom installation.	Decide which type installation is suitable for your environment:
	■ If you are using a graphical user interface (GUI) you can choose a default or a custom installation:
	  ■ A default installation formats the hard disk and installs a preselected set of software, including the Sun Java™ Enterprise System.
	  ■ A custom installation enables you to modify the hard disk layout and select the software that you want to install.
	■ If you use a text installer (non-graphical interface), you can select the default values or edit the values to select the software you want to install.
		For information about the Sun Java Enterprise System, see *Sun Java Enterprise System Technical Overview* at `http://docs.sun.com`

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 OS that you want to install. Determine the appropriate swap space layout for your system.	Chapter 2
Choose to install a system from local media or from the network.	Decide on the most appropriate installation media for your environment.	"Installing From the Network or From DVD or CDs?" on page 19

Gather information about your system.	■ For the Solaris installation program, complete the worksheet to collect all of the information that you need to install or upgrade.
	■ For the custom JumpStart installation method, decide which profile keywords to use in your profile. Then review the keyword descriptions to find the information about your system that you need.
		■ For the Solaris installation program, see either of the following documents:
		  ■ Chapter 3, Gathering Information Before Upgrade (Planning)
		  ■ Chapter 3.
		■ For the custom JumpStart installation method, see Chapter 9, "Custom JumpStart (Reference)," in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

TABLE 1–1 Task Map: Installing or Upgrading the Solaris Software *(Continued)*

Task	Description	For Instructions
(Optional) Set system parameters.	You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.	Chapter 4, "Preconfiguring System Configuration Information (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*.
(Optional) Prepare to install the Solaris software from the network.	If you chose to install the Solaris software from the network, create an installation 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, seeChapter 7, "Preparing to Install From the Network With CD Media (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*. To install over a wide area network, see Chapter 11, "Preparing to Install With WAN Boot (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*.
(Upgrade only) Perform tasks prior to upgrade.	Back up your system and determine if you can upgrade with disk space reallocation.	"Upgrade" on page 29.
Perform an installation 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 programs
Troubleshoot installation problems	Review the troubleshooting information when you encounter problems with your installation.	Appendix A.

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.

You can set up the systems to install from the network with remote DVD or CD images. You might want to set up systems this way for the following reasons:

- If you have systems that do not have local DVD-ROM or CD-ROM drives
- 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 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 "Choosing a Solaris Installation Method" on page 21.

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 about preparing to install from a local area network	Chapter 7, "Preparing to Install From the Network With CD Media (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*
For instructions about preparing to install over a wide area network	Chapter 11, "Preparing to Install With WAN Boot (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*
For instructions about how to install x86 based clients over the network by using PXE	"x86: Booting and Installing Over the Network With PXE" on page 209

Initial Installation or Upgrade?

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

Initial Installation

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

If the system is already running the Solaris OS, 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 "Choosing a Solaris Installation Method" on page 21.

Upgrade

You can upgrade the Solaris OS by using two upgrade methods: standard and Solaris Live Upgrade. A standard upgrade upgrades the OS by maintaining as many existing configuration parameters as possible of the current OS. Solaris Live Upgrade upgrades the OS by creating a copy of the current system. This copy can be upgraded with a standard upgrade. The upgraded OS can then be switched to become the current system by a simple reboot. If a failure occurs, you can switch back to the original OS with a reboot. Solaris Live Upgrade enables you to keep your system running while you upgrade and enables you to switch back and forth between OS releases.

You can upgrade any system that is running the Solaris 7, Solaris 8, or Solaris 9 software. For more information about upgrading and the list of upgrade methods, see "Upgrade" on page 29.

Choosing a Solaris Installation Method

The Solaris OS provides several programs for installation or upgrade. Each installation technology offers different features that are designed for specific installation requirements and environments. Use the following table to help you decide which installation method to use.

TABLE 1–2 Choosing Your Installation Method

Task	Installation Method	Instructions
Install one system from CD-ROM or DVD-ROM media with an interactive program.	Solaris installation program	*Solaris 10 Installation Guide: Basic Installations*
Install one system over a local area network.	Solaris installation program over the network	Part II, "Installing Over a Local Area Network," in *Solaris 10 Installation Guide: Network-Based Installations*
Automate the installation or upgrade of multiple systems based on profiles you create.	Custom JumpStart	Chapter 4, "Preparing Custom JumpStart Installations (Tasks)," in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*
Replicate the same software and configuration on multiple systems.	Solaris Flash archives	Chapter 1, "Solaris Flash (Overview)," in *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*

TABLE 1–2 Choosing Your Installation Method *(Continued)*

Task	Installation Method	Instructions
Install systems over a wide area network (WAN) or the Internet.	WAN boot	Chapter 9, "WAN Boot (Overview)," in *Solaris 10 Installation Guide: Network-Based Installations*
Upgrade a system while it is running.	Solaris Live Upgrade	Chapter 4
After installing the Solaris OS, create an isolated application environment.	Solaris Zones	Chapter 16, "Introduction to Solaris Zones," in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*

Sun Java System Application Server Platform Edition 8

The Sun Java System Application Server Platform Edition 8 provides for broad deployment of application services and web services. This software is automatically installed with the Solaris OS. You can find documentation for the server in the following areas:

- For documentation about starting the server:

  Sun Java System Application Server Platform Edition 8 QuickStart Guide. You can find this guide in the installation directory file `/docs/QuickStart.html`.
- For the full Application Server documentation set:

  `http://docs.sun.com/db/coll/ApplicationServer8_04q2`.
- For a tutorial:

  `http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html`.

CHAPTER 2

Solaris Installation and Upgrade (Planning)

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

System Requirements and Recommendations

TABLE 2–1 Memory, Swap, and Processor Recommendations

System	Size
Memory to install or upgrade	256 MB is the recommended size. 64 MB is the minimum size. **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 installation's text installer, not through the graphical user interface (GUI). For more information about these memory requirements, see Table 2–2.

TABLE 2–1 Memory, Swap, and Processor Recommendations *(Continued)*

System	Size
Swap area	512 Mbytes is the default size.
x86: Processor requirements	SPARC: 200–MHz or faster processor is required.
	x86: A 120–MHz or faster processor is recommended. Hardware floating-point support is required.

You can choose to install the software with a GUI or with or without a windowing environment. If there is sufficient memory, the GUI is displayed by default. Other environments are displayed by default if memory is insufficient for the GUI. You can override defaults with the `nowin` or `text` boot options. But, you are limited by the amount of memory in your system or by installing remotely. Also if the Solaris installation program does not detect a video adapter, it automatically displays in a console-based environment. Table 2–2 describes these environments and lists minimal memory requirements for displaying them.

TABLE 2–2 Minimal Memory Requirements for Display Options

Minimal Memory	Type of Installation	Description	Windowing Environment
64–127 MB	Console-based	Contains no graphics and no windowing environment. If no video adapter is detected, the installer displays a console-based environment. If you are installing remotely through a `tip` line or using the `nowin` boot option, you are limited to the console-based installation.	
128–383 MB	Console-based	Contains no graphics, but provides a window and the ability to open other windows. Requires a local or remote DVD-ROM or CD-ROM drive or network connection, video adapter, keyboard, monitor. If you install by using the `text` boot option and have enough memory, you are installing in a windowing environment.	X
384 MB or greater	GUI-based	Provides windows, pull-down menus, buttons, scrollbars, and iconic images. A GUI requires a local or remote DVD-ROM or CD-ROM drive or network connection, video adapter, keyboard, monitor.	X

Allocating Disk and Swap Space

Before you install the Solaris software, you can determine if your system has enough disk space by doing some high-level planning.

General Disk Space Planning and Recommendations

Planning disk space is different for everyone. Consider allocating space for the following conditions, depending on your needs.

TABLE 2–3 General Disk and Swap Space Planning

Conditions for Space Allocations	Description
File systems	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. By default, the Solaris installation methods create only root (/) and `/swap`. When space is allocated for OS services, the `/export` directory is also created. If you are upgrading to a major Solaris release, you might need to reslice your system or allocate double the space that you need at installation time. If you are upgrading to an update, you could prevent having to reslice your system by allocating extra disk space for future upgrades. A Solaris update release needs approximately 10 percent more disk space than the previous release. You can allocate an additional 30 percent of disk space for each file system to allow space for several Solaris updates.
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.

TABLE 2–3 General Disk and Swap Space Planning *(Continued)*

Conditions for Space Allocations	Description
Swap	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. 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 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 installation program, 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 about the Solaris Custom JumpStart profile file, see "Creating a Profile" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*. For an overview of the swap space, see Chapter 20, "Configuring Additional Swap Space (Tasks)," in *System Administration Guide: Devices and File Systems*.
A server that is providing home directory file systems	By default, home directories are usually located in the `/export` file system.
The Solaris software group you are installing	A software group is a grouping of software packages. When you are planning disk space, remember that you can add or remove individual software packages from the software group that you select. For information about software groups, see "Disk Space Recommendations for Software Groups" on page 26.
Upgrade	■ If you are using Solaris Live Upgrade to upgrade an inactive boot environment and want information about disk space planning, see "Solaris Live Upgrade Disk Space Requirements" on page 72. ■ If you are using other Solaris installation methods to plan disk space, see "Upgrading With Disk Space Reallocation" on page 31.
Supporting languages	For example, Chinese, Japanese, or Korean.
Supporting printing or mail	Allocate additional space.
Additional software or third-party software	Allocate additional space.

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.

- For an initial installation, you select the software group to install, based on the functions that you want to perform on the system.

- For an upgrade, you must upgrade to a software group that is 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.

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 figure shows the grouping of software packages. Reduced Network Support contains the minimal number of packages and Entire Solaris Software Group Plus OEM Support contains all the packages.

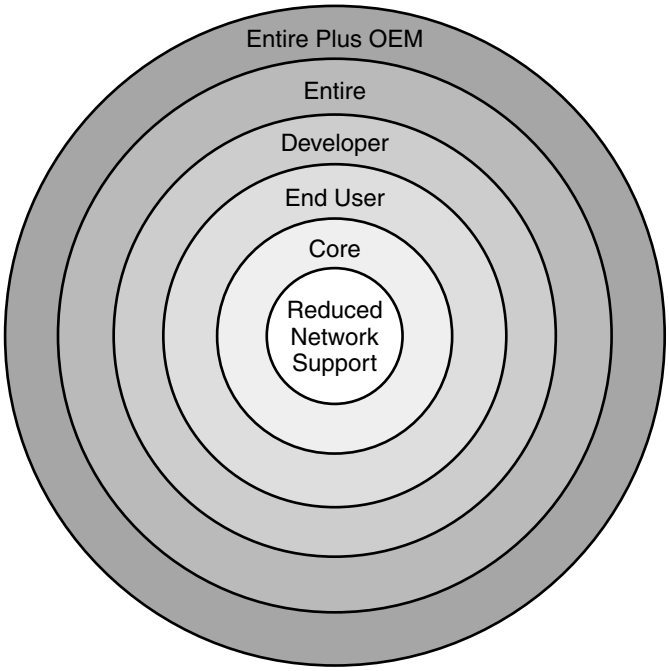


FIGURE 2–1 Solaris Software Groups

Table 2–4 lists the Solaris software groups and the recommended amount of disk space that you need to install each group.

Note – The disk space recommendations in Table 2–4 include space for the following items.

- Swap space
- Patches
- Additional software packages

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

TABLE 2–4 Disk Space Recommendations for Software Groups

Software Group	Description	Recommended Disk Space
Entire Solaris Software Group Plus OEM Support	Contains the packages for the Entire Solaris Software Group plus additional hardware drivers, including drivers for hardware that is not on the system at the time of installation.	6.7 Gbytes
Entire Solaris Software Group	Contains the packages for the Developer Solaris Software Group and additional software that is needed for servers.	6.5 Gbytes
Developer Solaris Software Group	Contains the packages for the End User Solaris 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.	6.0 Gbytes
End User Solaris Software Group	Contains the packages that provide the minimum code that is required to boot and run a networked Solaris system and the Common Desktop Environment.	5.0 Gbytes
Core System Support Software Group	Contains the packages that provide the minimum code that is required to boot and run a networked Solaris system.	2.0 Gbytes
Reduced Network Support Software Group	Contains the packages that provide the minimum code that is required to boot and run a Solaris system with limited network service support. The Reduced Networking Software Group provides a multiuser text-based console and system administration utilities. This software group also enables the system to recognize network interfaces, but does not activate network services.	2.0 Gbytes

Upgrade

You can upgrade a system by using one of three different upgrade methods: Solaris Live Upgrade, the Solaris installation program, and custom JumpStart.

TABLE 2–5 Solaris Upgrade Methods

Current Solaris OS	Solaris Upgrade Methods
Solaris 7, Solaris 8, Solaris 9	■ Solaris Live Upgrade – Upgrades a system by creating and upgrading a copy of the running system ■ The Solaris installation – Provides an interactive upgrade with a graphical user interface or command-line interface ■ Custom JumpStart method – Provides an automated upgrade

Upgrade Limitations

Issue	Description
Upgrading to a different 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.
Using the Solaris installation program to upgrade from a CD or DVD	You must have a free slice on the disk that does not store files and can be overwritten by the installation software. 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.

Upgrade Programs

You can perform a standard interactive upgrade with the Solaris installation program or an unattended upgrade with the custom JumpStart installation method. Solaris Live Upgrade enables you to upgrade a running system.

Upgrade Program	Description	For More Information
Solaris Live Upgrade	Enables you to create a copy of the currently running system. The copy can be upgraded and then a reboot switches the upgraded copy to become the currently running system. Using Solaris Live Upgrade reduces the downtime that is required to upgrade the OS. Also, Solaris Live Upgrade can prevent problems with upgrading such as not being able to recover from an upgrade if the power fails, because the copy being upgraded is not the currently running system.	To plan for disk space allocation when using Solaris Live Upgrade, see Chapter 5.
The Solaris installation program	Guides you through an upgrade with a GUI.	Chapter 2, "Installing With the Solaris Installation Program (Tasks)," in *Solaris 10 Installation Guide: Basic Installations*.
Custom JumpStart program	Provides an automated upgrade. A profile file and optional preinstallation and postinstallation scripts provide the information required. When creating a custom JumpStart profile for an upgrade, 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 with 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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*. ■ For more information about creating a upgrade profile, see "Profile Examples" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations* ■ For more information about performing an upgrade, see "Performing a Custom JumpStart Installation" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*

Installing a Solaris Flash Archive Instead of Upgrading

The Solaris Flash installation feature provides a method of creating a copy of the whole installation from a master system that can be replicated on many clone systems. This copy is called a Solaris Flash archive. You can install an archive by using any installation program. For information about installing an archive, see the following table.

Solaris Live Upgrade	"Installing Solaris Flash Archives on a Boot Environment" on page 135
Custom JumpStart	"To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*
Solaris installation program	Chapter 4, "Installing and Administering Solaris Flash Archives (Tasks)," in *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*
WAN boot installation method	Chapter 12, "Installing With WAN Boot (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*

Upgrading With Disk Space Reallocation

The upgrade option in the Solaris installation program and the `upgrade` keyword in the custom JumpStart program provide the ability to reallocate disk space. This reallocation automatically changes the sizes of the disk slices. You can reallocate disk space if the current file systems do not have enough space for the upgrade. For example, 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 installation program, and auto-layout cannot determine how to reallocate the disk space, you must use the custom JumpStart program to upgrade.
- 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 "Profile Examples" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Using the Solaris Installation Program to Upgrade From DVD or CD Media

When you use the Solaris installation program from a DVD or CD 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 the `/etc/vfstab`. The size of this slice must be at least 512 Mbytes.

Backing Up Systems Before Upgrading

Backing up your existing file systems before you upgrade to the Solaris OS is highly recommended. 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 Chapter 23, "Backing Up and Restoring File Systems (Overview)," in *System Administration Guide: Devices and File Systems*.

How to Find the Version of the OS That Your System Is Running

To see the version of Solaris software that is running on your system, type either of the following commands.

```
$ uname -a
```

The `cat` command provides more detailed information.

```
$ cat /etc/release
```

Locale Values

As a part of your installation, you can preconfigure the locale that you want the system to use. A *locale* determines how online information is displayed in a specific language and specific 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.

You can preconfigure the system locale in a custom JumpStart profile or in the `sysidcfg` file.

Setting the locale in a profile	"Creating a Profile" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*
Setting the locale in the `sysidcfg` file	"Preconfiguring With the sysidcfg File" in *Solaris 10 Installation Guide: Network-Based Installations*
List of locale values	*International Language Environments Guide*

Platform Names and Groups

You must know your system architecture (platform group) when 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 10 Sun Hardware Platform Guide* at `http://docs.sun.com/`.

TABLE 2–6 Example of Platform Names and Groups

System	Platform Name	Platform Group
Sun Blade™	SUNW,Sun-Blade-100	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*.

Planning to Install and Configure Zones

The following introduction provides high-level planning information for global and non-global zones. For more specific planning information and specific procedures, see Chapter 16, "Introduction to Solaris Zones," in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

After the Solaris OS is installed, you can install and configure zones. In a zones environment, the global zone is the single instance of the operating system that is running and is contained on every Solaris system. The global zone is both the default zone for the system and the zone that is used for system-wide administrative control. A non-global zone is a virtualized operating system environment.

Solaris Zones are a software partitioning technology used to virtualize operating system services and provide an isolated and secure environment for running applications. When you create a zone, you produce an application execution environment in which processes are isolated from all other zones. This isolation prevents processes that are running in one zone from monitoring or affecting processes that are running in any other zones. Even a process running in a non-global zone with superuser credentials cannot view or affect activity in any other zones. A process running in the global zone with superuser credentials can affect any process in any zone.

The global zone is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled. Only the global zone is bootable from the system hardware. Administration of the system infrastructure, such as physical devices, routing, or dynamic reconfiguration (DR), is only possible in the global zone. Appropriately privileged processes running in the global zone can access objects associated with any or all other zones.

When installing the OS, the software group installed in the global zone is the set of packages that is shared by all the non-global zones. For example, if you install the Entire software group, all zones contain these packages. By default any additional packages installed in the global zone also populate the non-global zones. You can segregate into non-global zones applications, namespaces, servers, and network connections such as NFS and DHCP as well as other software. Each non-global zone is not aware of nor does it contain the segregated software in other non-global zones and each can operate independently. For example, you might have installed the Entire

software group on the global zone and have running on separate non-global zones the Java Enterprise System Messaging Server, a database, DHCP, and a web server. When installing non-global zones keep in mind performance requirements of the applications running in each non-global zone.

Disk Space Requirements for Non-Global Zones

When installing the global zone, be sure to reserve enough disk space to house all of the zones you might create. Each non-global zone might have unique disk space requirements. Below is a brief overview of planning information. For complete planning requirements and recommendations, see Chapter 18, "Planning and Configuring Non-Global Zones (Tasks)," in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

No limits are placed on how much disk space can be consumed by a zone. The global zone administrator is responsible for space restriction. Even a small uniprocessor system can support a number of zones running simultaneously.

The nature of the packages installed in the global zone affects the space requirements of the non-global zones that are created. The number of packages and space requirements are factors. The following are general disk space guidelines.

- Approximately 100 Mbytes of free disk space is suggested when the global zone has been installed with all of the standard Solaris packages. Increase this amount if additional packages are installed in the global zone. By default, any additional packages installed in the global zone also populate the non-global zones. The directory location in the non-global zone for these additional packages is specified through the `inherit-pkg-dir` resource.
- Add 40 Mbytes of RAM per zone if the system has sufficient swap space. This addition is recommended to make each zone operational. When planning your system size, take this addition of RAM into consideration.

Restricting Non-Global Zone Size

The following options can be used to restrict zone size.

- You can place the zone on a `lofi`-mounted partition. This action limits the amount of space consumed by the zone to that of the file used by `lofi`. For more information, see the `lofiadm`(1M) and `lofi`(7D) man pages.
- You can use soft partitions to divide disk slices or logical volumes into partitions. You can use these partitions as zone roots, and thus limit per-zone disk consumption. The soft partition limit is 8192 partitions. For more information, see Chapter 12, "Soft Partitions (Overview)," in *Solaris Volume Manager Administration Guide*.
- You can use the standard partitions of a disk for zone roots, and thus limit per-zone disk consumption.

SPARC: 64–bit Packaging Changes

In previous Solaris releases, the Solaris OS was delivered in separate packages for 32-bit and 64-bit components. In the Solaris 10 OS, packaging has been simplified with most 32-bit and 64-bit components being delivered in a single package. The combined packages retain the names of the original 32-bit packages, and the 64-bit packages are no longer delivered. This change reduces the number of packages and simplifies installation.

The 64-bit packages are renamed with the following conventions:

- If a 64-bit package has a 32-bit counterpart, the 64-bit package is named with the 32-bit package name. For example, a 64-bit library such as /usr/lib/sparcv9/libc.so.1 previously would have been delivered in SUNWcslx, but now is delivered in SUNWcsl. The 64-bit SUNWcslx package is no longer delivered.
- If a package does not have a 32-bit counterpart, the "x" suffix is removed from the name. For example, SUNW1394x becomes SUNW1394. This change means that you might need to modify your custom JumpStart script or other package installation scripts to remove references to the 64-bit packages.

x86: Partitioning Recommendations

When using the x86 OS, follow these guidelines for partitioning your system.

The Solaris installation uses a default boot-disk partition layout. These partitions are called `fdisk` partitions. An fdisk partition is 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 `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.

TABLE 2–7 x86: Default Partitions

Partitions	Partition Name	Partition Size
First partition (on some systems)	Diagnostic or Service partition	Existing size on system

TABLE 2–7 x86: Default Partitions *(Continued)*

Partitions	Partition Name	Partition Size
Second partition	x86 boot partition	Greater than 10 Mbytes, depending on disk size
Third partition	Solaris OS partition	Remaining space on the boot disk

Default Boot-Disk Partition Layout Preserves the Service Partition

The Solaris installation program uses a default boot-disk partition layout to accommodate the diagnostic or Service partition. If your system currently includes a diagnostic or Service partition, the default boot-disk partition layout enables you to preserve this partition.

Note – If you install the Solaris OS on an x86 based system that does not currently include a diagnostic or Service partition, the installation program does not create a new diagnostic or Service partition by default. If you want to create a diagnostic or Service partition on your system, see your hardware documentation.

CHAPTER 3

Gathering Information Before Upgrade (Planning)

This chapter contains a worksheet to help you gather all of the information that you need to upgrade your system.

Checklist for Upgrading

Use the following checklist to gather the information that you need for a standard upgrade of the Solaris OS. You do not need to gather all of the information that is requested on the checklist. 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 installation program to upgrade, the upgrade fails if you attempt to change any of the values.

TABLE 3–1 Upgrade Checklist

Information for Upgrade	Description or Example	Answer – Defaults are noted with an asterisk (*)
Network connection	Is the system connected to a network?	Networked/Nonnetworked*
DHCP	Can the system use Dynamic Host Configuration Protocol (DHCP) to configure its network interfaces?	Yes/No*

TABLE 3–1 Upgrade Checklist *(Continued)*

Information for Upgrade		Description or Example	Answer – Defaults are noted with an asterisk (*)
If you are not using DHCP, note the network address.	IP Address	If you are not using DHCP, supply the IP address for the system. Example: 172.31.255.255 To find this information about a running system, type the following command. # **ypmatch** *host-name* **hosts**	
	Subnet	If you are not using DHCP, is the system part of a subnet? If yes, what is the netmask of the subnet? Example: 255.255.255.0 To find this information about a running system, type the following command. # **more /etc/netmasks**	255.255.255.0*
	IPv6	Do you want to enable IPv6 on this machine?	Yes/No*
Host Name.		Host name that you choose for the system. To find this information about a running system, type the following command. # **uname -n**	
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*
If the system uses a name service, provide the following information.	Name Service	Which name service should this system use? To find this information about a running system, type the following command. # **cat /etc/nsswitch.conf**	NIS+/NIS/DNS/ LDAP/None*

TABLE 3–1 Upgrade Checklist *(Continued)*

Information for Upgrade		Description or Example	Answer – Defaults are noted with an asterisk (*)
	Domain Name	Provide the name of the domain in which the system resides. To find this information about a running system, type the following command. # **domainname**	
	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: To display the server's host name, type the following command. # **ypwhich** Server's IP Address: To display the server's IP address, type the following command. # **nismatch** *nameserver-name* **hosts.org_dir**	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: To display the server's IP address, type the following command. # **getents ipnodes dns** You can enter a list of domains to search when a DNS query is made. Search Domain: Search Domain: Search Domain:	
	LDAP	Provide the following information about your LDAP profile. Profile Name: Profile Server:	

TABLE 3–1 Upgrade Checklist *(Continued)*

Information for Upgrade		Description or Example	Answer – Defaults are noted with an asterisk (*)
		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 installation 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 Solaris installation program detect an IP address. However, the system must be on a subnet that has a router that advertises itself by using the ICMP router discovery protocol. If you are using the command-line interface, the software detects an IP address when the system is booted. ■ 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.	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
Root Password		Provide the root password for the system.	

TABLE 3–1 Upgrade Checklist *(Continued)*

Information for Upgrade	Description or Example	Answer – Defaults are noted with an asterisk (*)
Default or Custom Install	Do you want to perform a default installation, or customize the installation? ■ Select Default installation to format the entire hard disk and install a preselected set of software, including Sun Java Enterprise System. For more information, see Sun Java Enterprise System Technical Overview on http://docs.sun.com. ■ Select Custom installation to modify the hard disk layout and select the software that you want to install. **Note –** The text installer does not prompt you to select a Default or Custom Installation. To perform a default installation, accept the default values that are provided in the text installer. To perform a custom installation, edit the values in the text installer screens.	Default installation*/Custom installation
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.	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

TABLE 3–1 Upgrade Checklist *(Continued)*

Information for Upgrade	Description or Example	Answer – Defaults are noted with an asterisk (*)
Disk space reallocation.	Do you want the installation program to automatically re-layout the systems on your disks? If yes, which file system should be used for auto-layout? Example: /, /opt, /var If no, you must provide information for the system configuration.	Yes/No*
If you are installing through a `tip` line, follow these instructions.	Ensure that your window display is at least 80 columns wide and 24 rows long. For more information, see `tip`(1). To determine the current dimensions of your `tip` window, use the `stty` command. For more information, see the man page, `stty`(1).	
Check your Ethernet connection.	If the system is part of a network, verify that an Ethernet connector or similar network adapter is connected to your system.	
Solaris Live Upgrade use.	■ Determine your resource requirements for creating a new boot environment and upgrading it. For detailed information, refer to Chapter 5. ■ Determine requirements if you are using RAID-1 volumes. For detailed information, refer to "Guidelines for Selecting Slices for File Systems" on page 77.	
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 and you're using Prestoserve software, you might lose data. Refer to the Prestoserve documentation for shutdown instructions.	
Check for patches needed.	The most recent patch list is provided at http://sunsolve.sun.com.	

TABLE 3–1 Upgrade Checklist *(Continued)*

Information for Upgrade	Description or Example	Answer – Defaults are noted with an asterisk (*)
Review the planning chapter and other relevant documentation.	■ Review the entire planning chapter or specific sections in Chapter 2. ■ Review the *Solaris 10 Release Notes* and vendor release notes to ensure that the software you use is supported in the new Solaris release. ■ Review the *Solaris 10 Sun Hardware Platform Guide* to ensure that your hardware is supported. ■ Review the documentation that accompanied your system to ensure that your system and devices are supported by the Solaris release.	

PART II Upgrading With Solaris Live Upgrade

This part provides an overview and instructions for using Solaris Live Upgrade to create and upgrade an inactive boot environment. The boot environment can then be switched to become the current boot environment.

CHAPTER 4

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 about 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 overview describes 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. The fallback process of switching back to the original boot environment is also described. Figure 4–1 describes this complete Solaris Live Upgrade process.

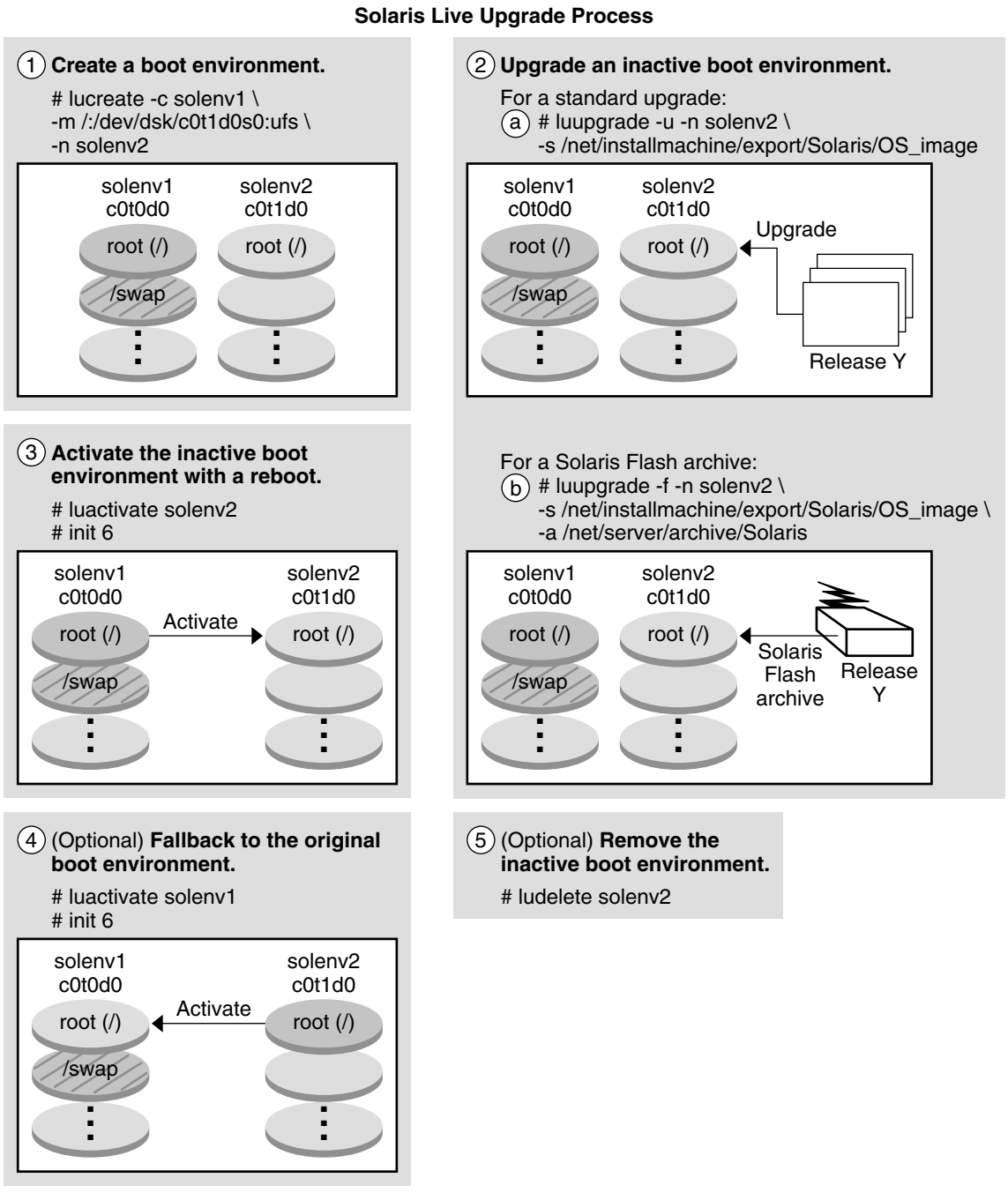


FIGURE 4–1 Solaris Live Upgrade Process

The following sections describe the Solaris Live Upgrade process.

1. A new boot environment can be created on a physical slice or a logical volume:
   - "Creating a Boot Environment" on page 52
   - "Creating a Boot Environment With RAID-1 Volume File Systems" on page 57
2. "Upgrading a Boot Environment" on page 64
3. "Activating a Boot Environment" on page 67
4. "Falling Back to the Original Boot Environment" on page 69

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. The following table describes these file system types.

File System Type	Description	Examples and More Information
Critical file systems	Critical file systems are required by the Solaris OS. These file systems are separate mount points in the `vfstab` of the active and inactive boot environments. These file systems are always copied from the source to the inactive boot environment. Critical file systems are sometimes referred to as *nonshareable*.	Examples are root (`/`), `/usr`, `/var`, or `/opt`.
Shareable file systems	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.	`/export` is an example of a file system that can be shared. For more detailed information about shareable file systems, see "Guidelines for Selecting Slices for Shareable File Systems" on page 81.

File System Type	Description	Examples and More Information
Swap	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 about reconfiguring swap, see the following: ■ "To Create a Boot Environment (Character Interface)" Step 9 ■ "To Create a Boot Environment and Reconfiguring Swap (Command-Line Interface)" on page 104

Creating RAID-1 Volumes on File Systems

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 RAID-1 Volume File Systems" on page 57.

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 to split or merging file systems	■ "To Create a Boot Environment (Character Interface)" Step 7 or Step 8 ■ "To Create a Boot Environment and Merge File Systems (Command-Line Interface)" on page 100 ■ "To Create a Boot Environment and Split File Systems (Command-Line Interface)" on page 102
For an overview of creating a boot environment with RAID–1 volume file systems	"Creating a Boot Environment With RAID-1 Volume File Systems" on page 57

Examples of Creating a New Boot Environment

The following figures illustrate various ways of creating new boot environments.

Figure 4–2 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.

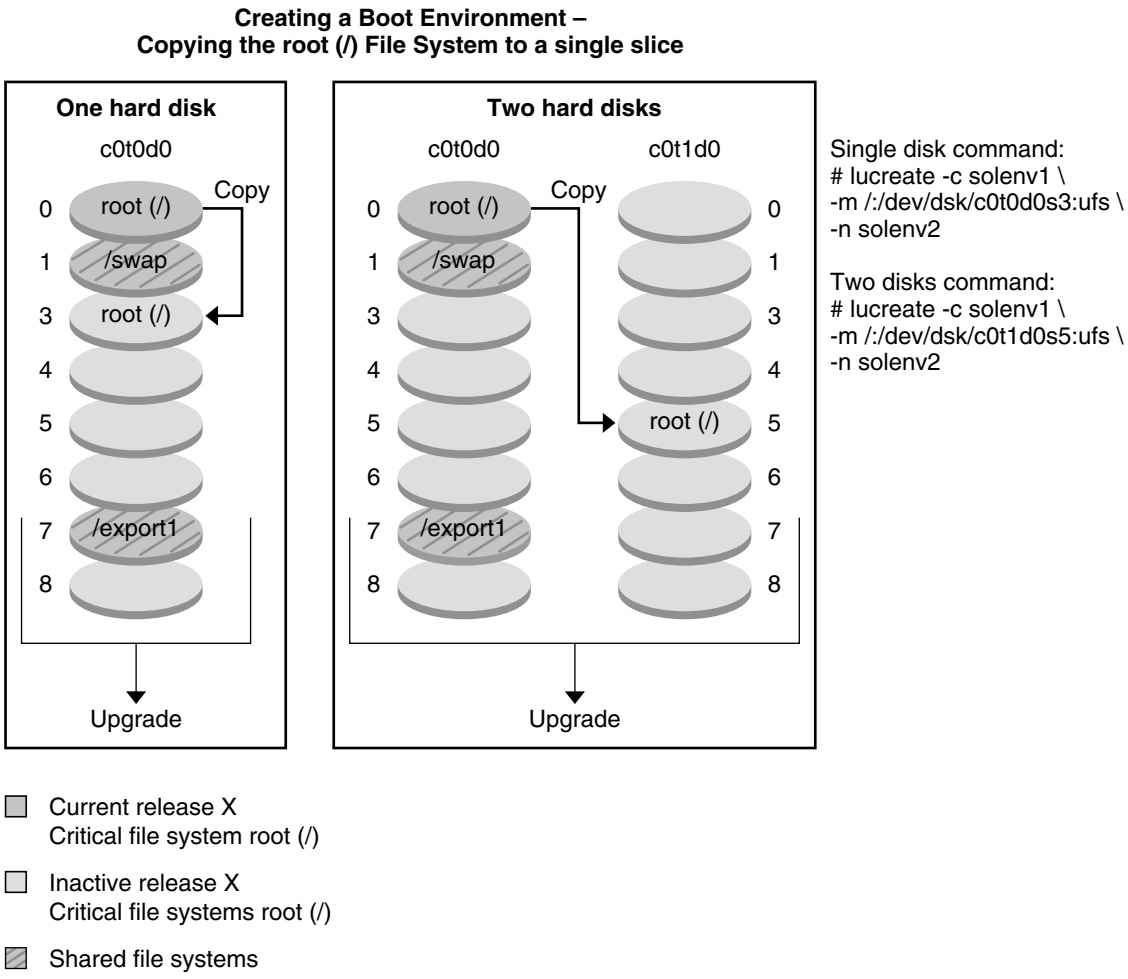


FIGURE 4–2 Creating an Inactive Boot Environment – Copying the root (/) File System

Figure 4–3 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.

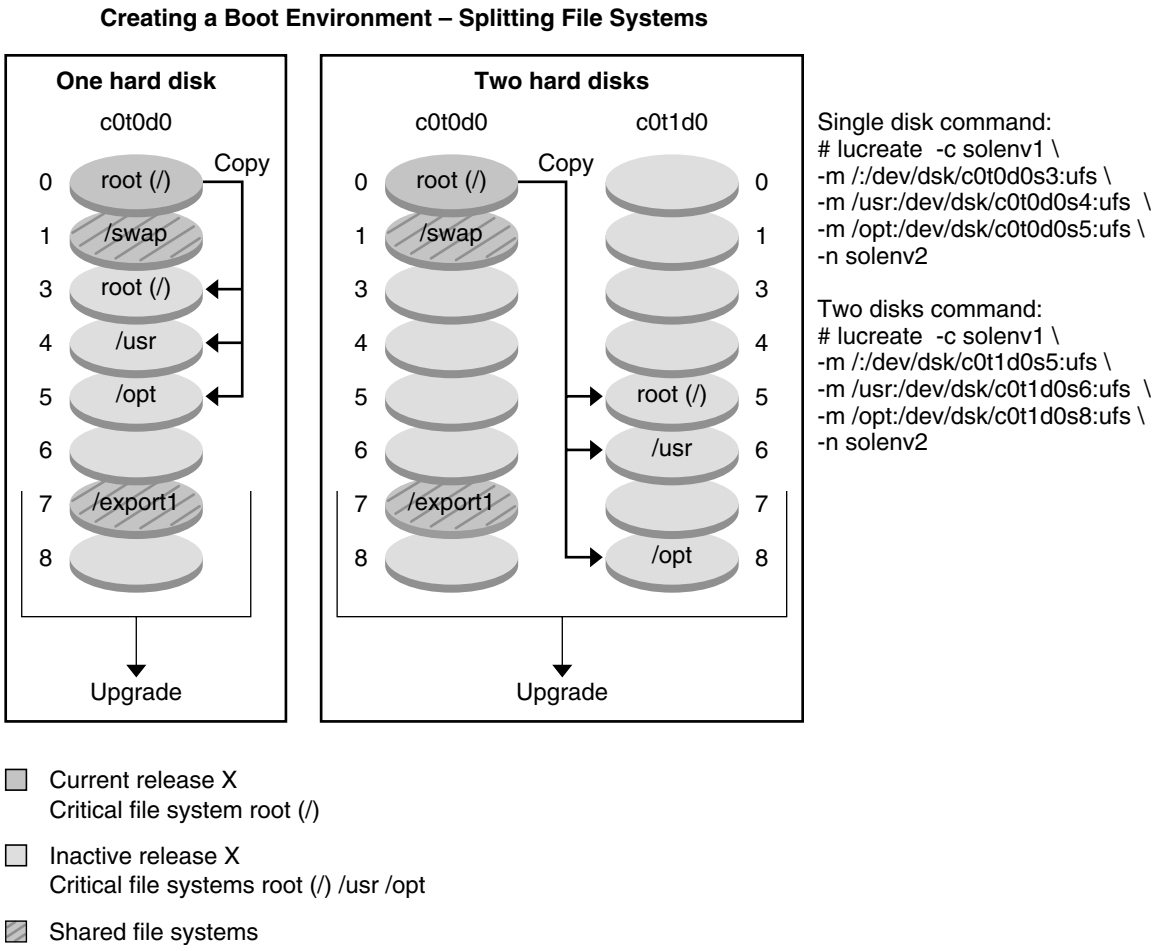


FIGURE 4–3 Creating an Inactive Boot Environment – Splitting File Systems

Figure 4–4 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.

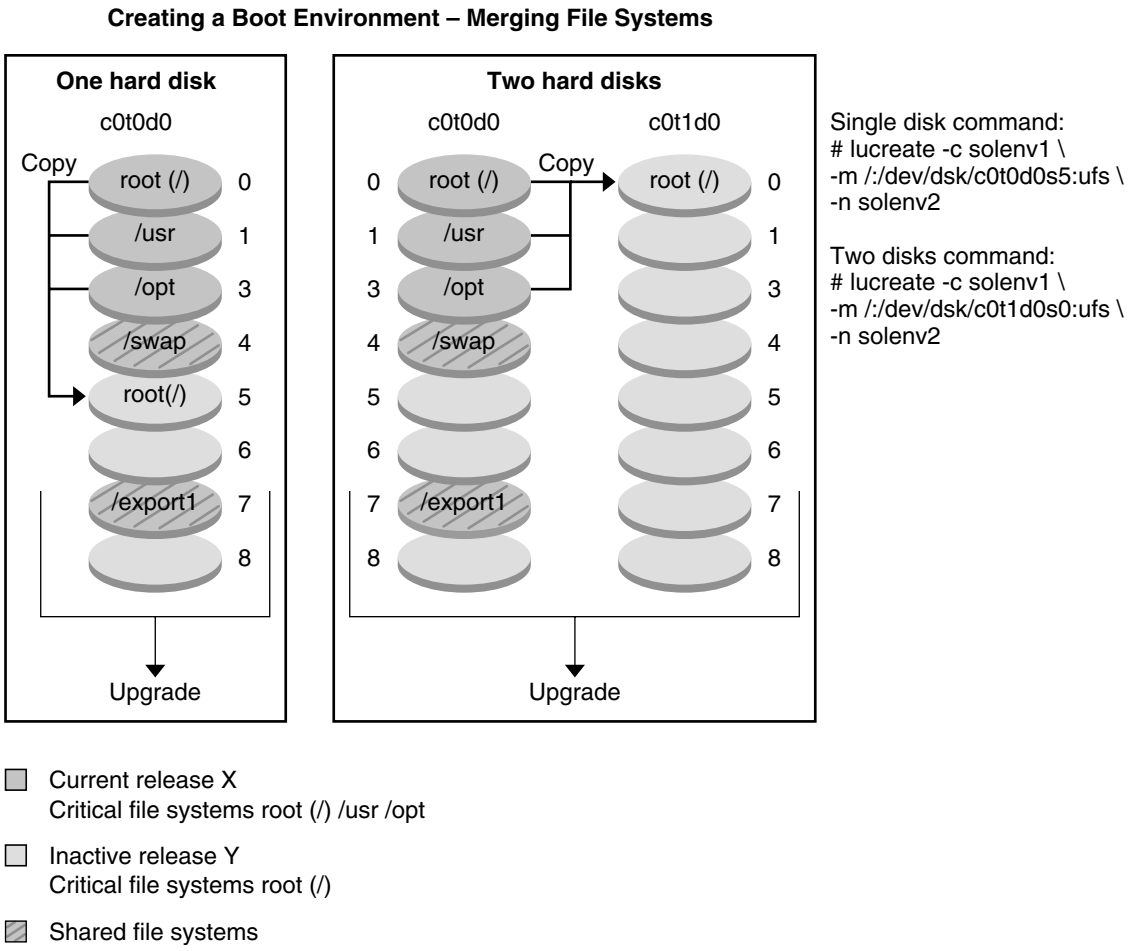


FIGURE 4–4 Creating an Inactive Boot Environment – Merging File Systems

Creating a Boot Environment With RAID-1 Volume File Systems

Solaris Live Upgrade uses Solaris Volume Manager technology to create a boot environment that can contain file systems encapsulated in RAID-1 volumes. 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 OS. 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.

How to Manage Volumes With Solaris Live Upgrade

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.

You use the `lucreate` command with the `-m` option to create a mirror, detach submirrors, and attach submirrors for the new boot environment.

For step-by-step procedures	"To Create a Boot Environment With RAID-1 Volumes (Mirrors) (Command-Line Interface)" on page 113
For an overview of creating RAID-1 volumes when installing	Chapter 10, "Creating RAID-1 Volumes (Mirrors) During Installation (Overview)," in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*
For in-depth information about other complex Solaris Volume Manager configurations that are not supported if you are using Solaris Live Upgrade	Chapter 2, "Storage Management Concepts," in *Solaris Volume Manager Administration Guide*

Mapping Solaris Volume Manager Tasks to Solaris Live Upgrade

Solaris Live Upgrade manages a subset of Solaris Volume Manager tasks. Table 4–1 shows the Solaris Volume Manager components that Solaris Live Upgrade can manage.

TABLE 4–1 Classes of Volumes

Term	Description
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.
mirror	A RAID-1 volume. See RAID-1 volume.
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 block for mirrors.
state database	A state database stores information about 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.
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.

Examples of Using Solaris Live Upgrade to Create RAID-1 Volumes

The following examples present command syntax for creating RAID-1 volumes for a new boot environment.

Create RAID-1 Volume on Two Physical Disks

Figure 4–5 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.

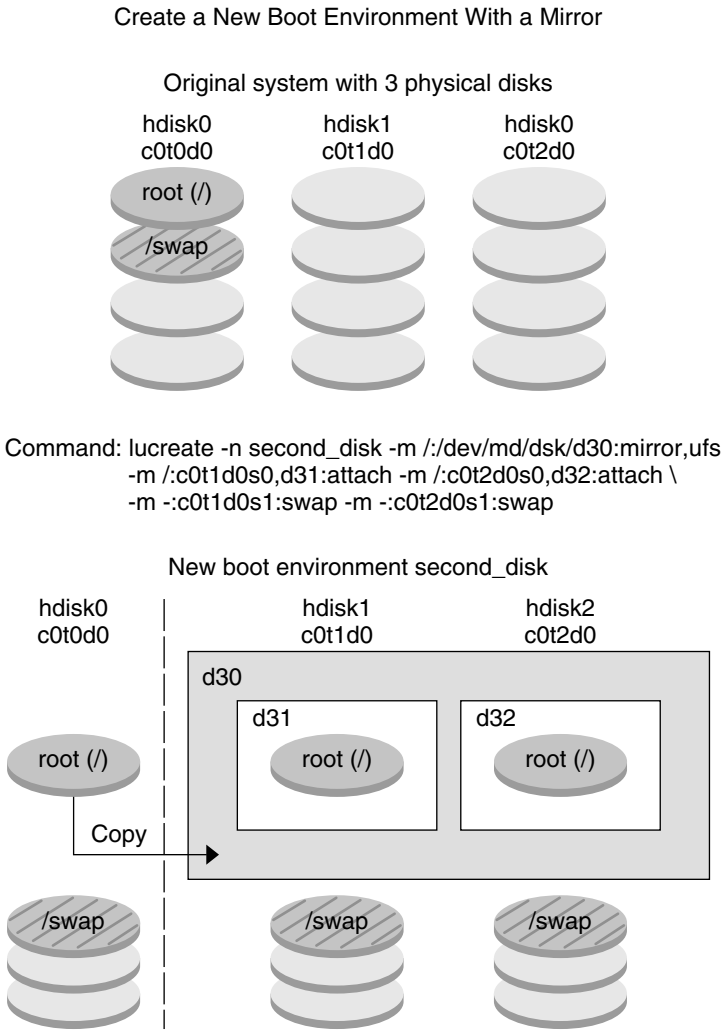


FIGURE 4–5 Create a Boot Environment and Create a Mirror

Create a Boot Environment and Use the Existing Submirror

Figure 4–6 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`. 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`.

Create a New Boot Environment and Use the Existing Submirror

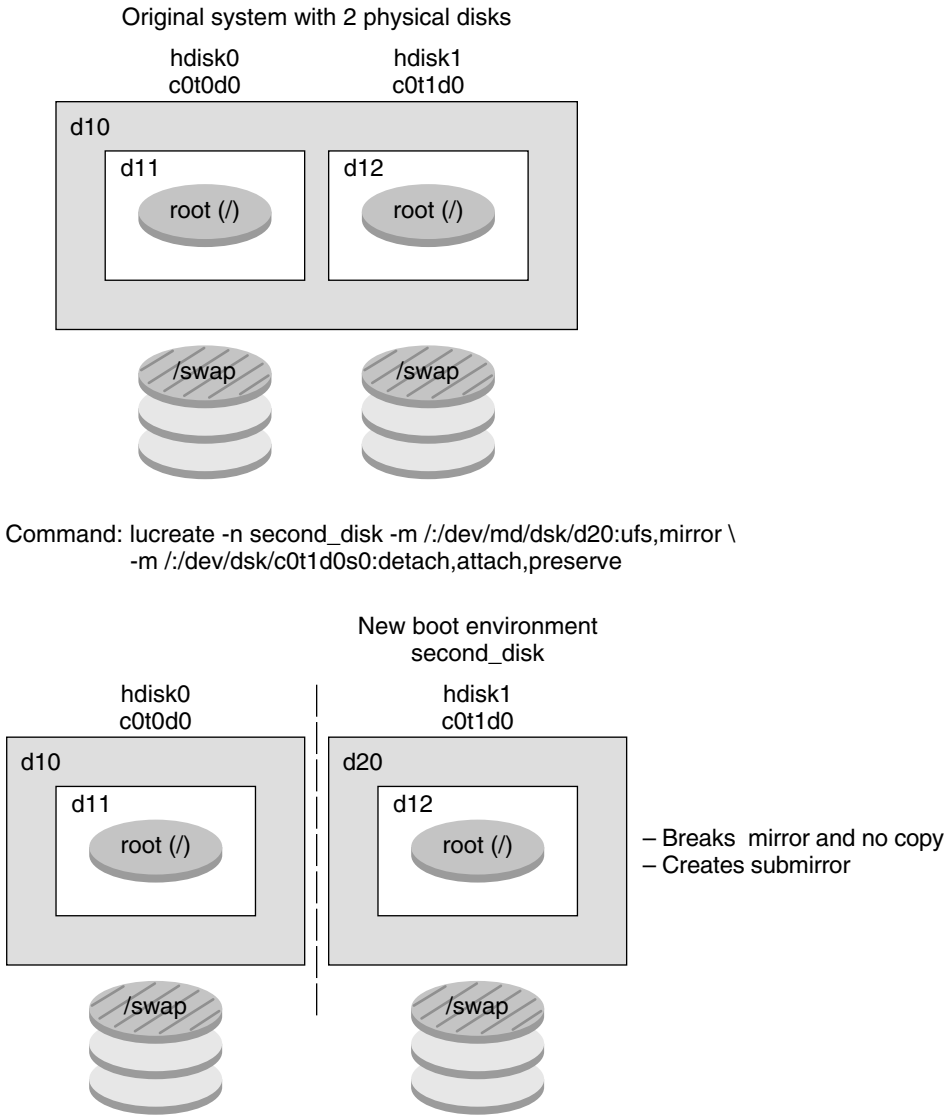


FIGURE 4–6 Create a Boot Environment and Use the Existing Submirror

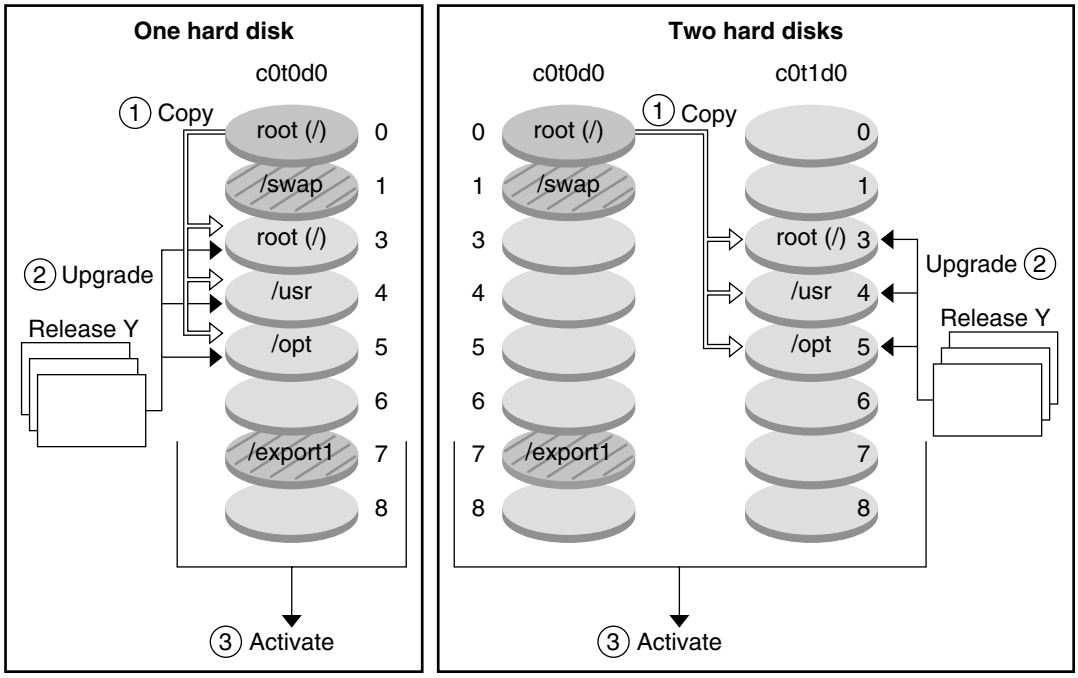
Upgrading a Boot Environment

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	Chapter 7
For an example of upgrading a boot environment with a RAID–1 volume file system	"Example of Detaching and Upgrading One Side of a RAID-1 Volume (Mirror) (Command-Line Interface)" on page 170

Figure 4–7 shows an upgrade to an inactive boot environment.

Upgrading a Boot Environment



Current release X
Critical file system root (/)

Inactive release Y
Critical file systems root (/) /usr /opt

Shared file systems

1 Single disk command:

```
# lucreate -c solenv1 \
-m /:/dev/dsk/c0t0d0s3:ufs -m /usr:/dev/dsk/c0t0d0s4:ufs \
-m /opt:/dev/dsk/c0t0d0s5:ufs \
-n solenv2
```

1 Two disks command:

```
# lucreate -c solenv1 \
-m /:/dev/dsk/c0t1d0s3:ufs -m /usr:/dev/dsk/c0t1d0s4:ufs \
-m /opt:/dev/dsk/c0t1d0s5:ufs
-n solenv2
```

2

```
# luupgrade -u -n solenv2 \
-s /net/installmachine/export/Solaris_10/OS_image
```

FIGURE 4–7 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 OS 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 about installing a Solaris Flash archive, see "Installing Solaris Flash Archives on a Boot Environment" on page 135.

The following figures show an installation of a Solaris Flash archive on an inactive boot environment. Figure 4–8 shows a system with a single hard disk. Figure 4–9 shows a system with two hard disks.

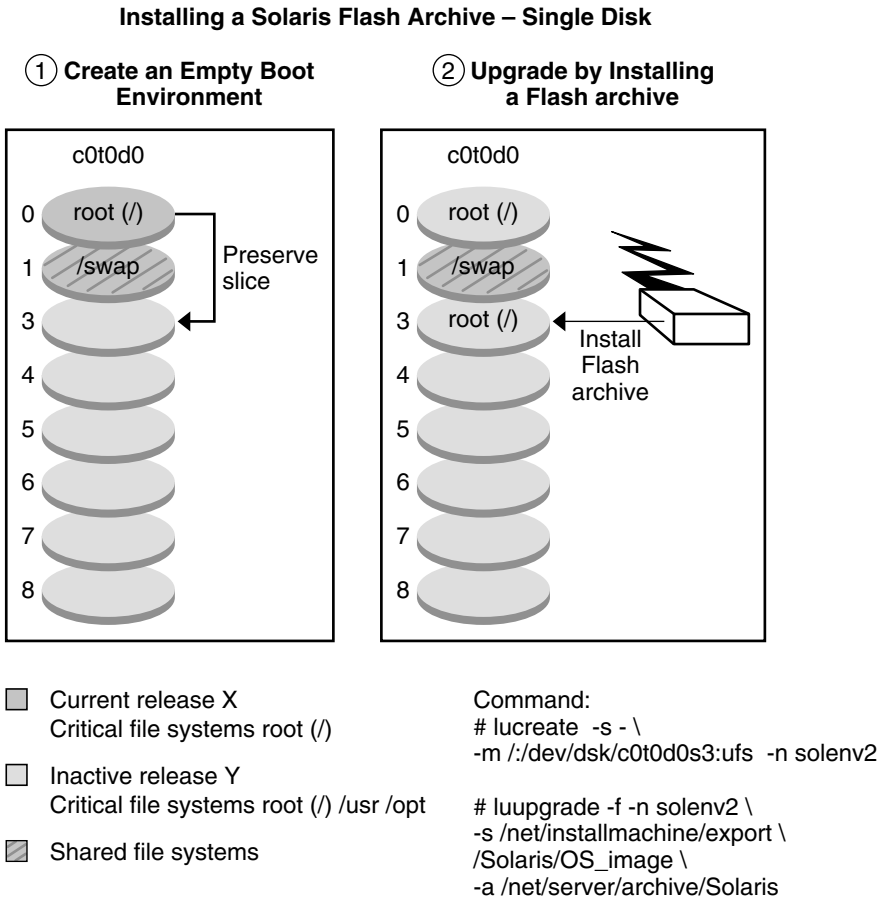


FIGURE 4–8 Installing a Solaris Flash Archive on a Single Disk

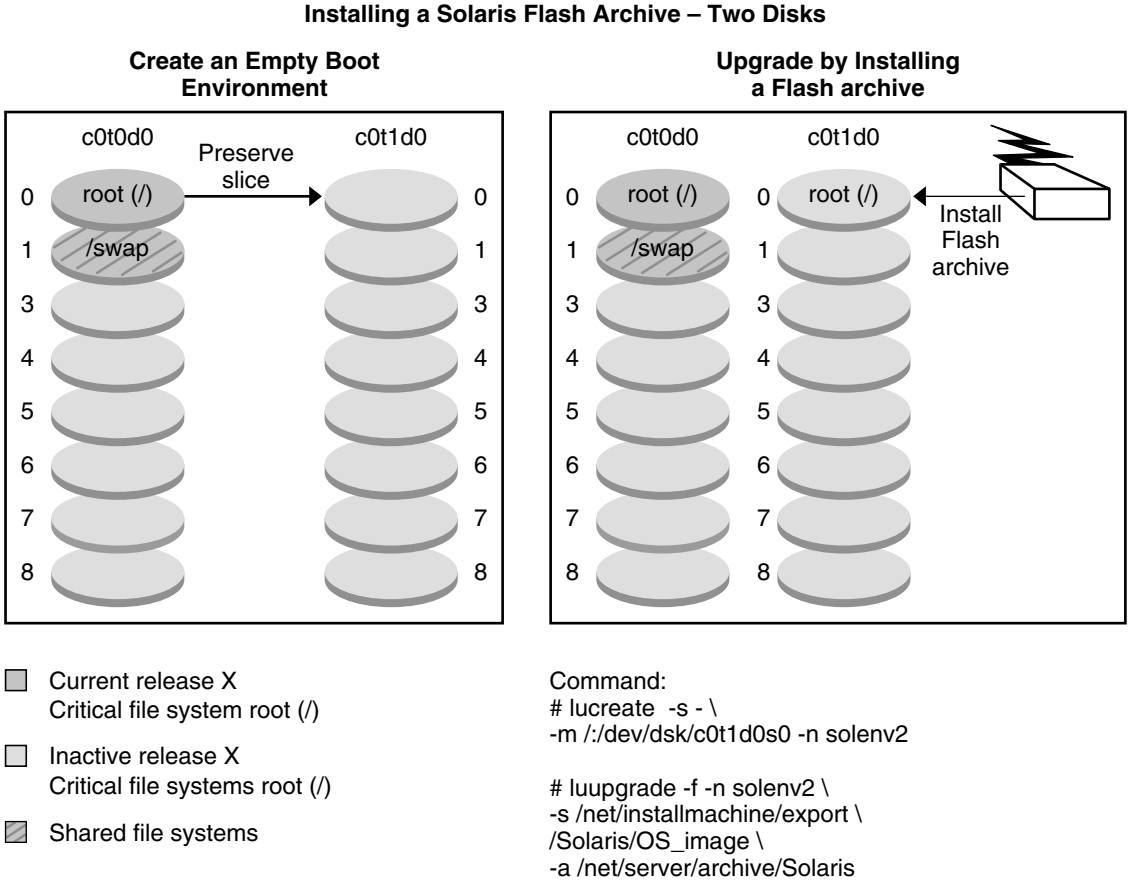


FIGURE 4–9 Installing a Solaris Flash Archive on Two Disks

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	"Activating a Boot Environment" on page 140
For information about synchronizing the active and inactive boot environment	"Synchronizing Files Between Boot Environments" on page 82

Figure 4–10 shows a switch after a reboot from an inactive to an active boot environment.

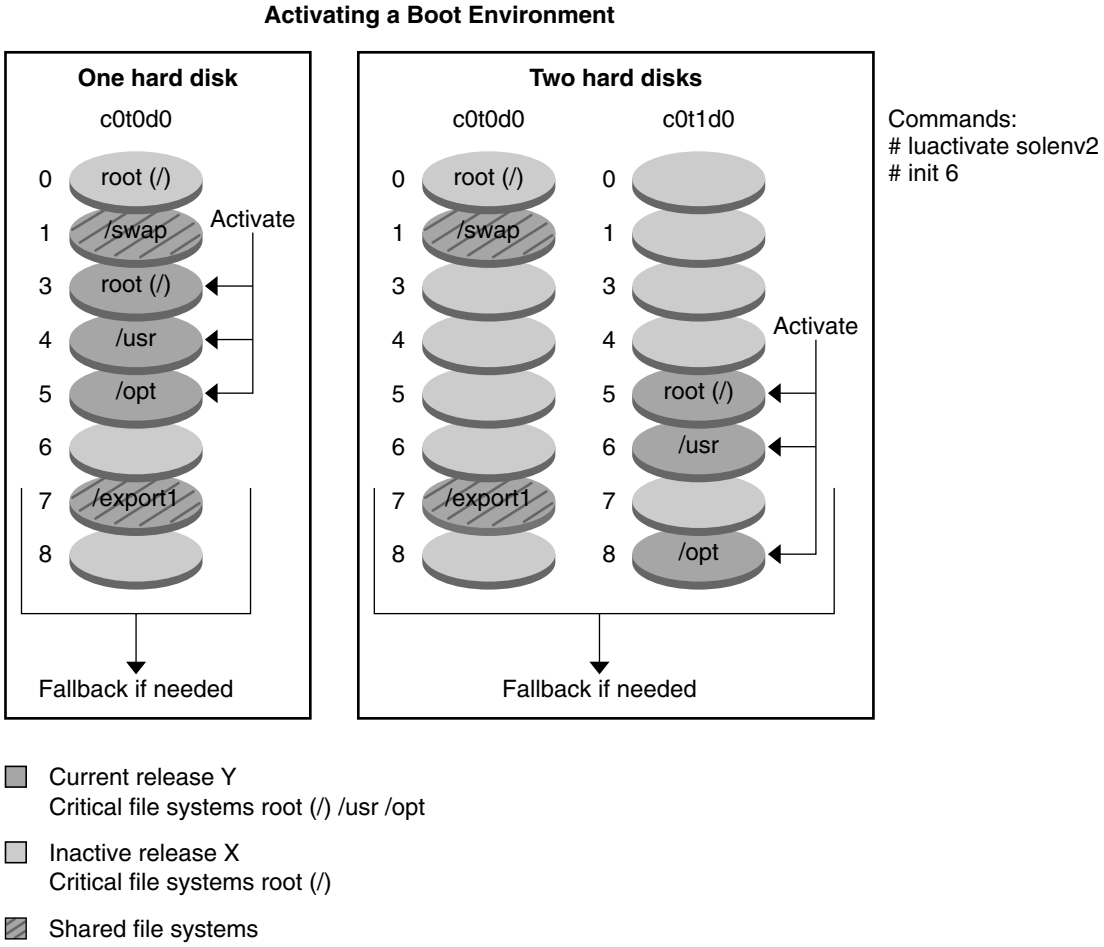


FIGURE 4–10 Activating an Inactive Boot Environment

Falling 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. 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:

Problem	Action
If the new boot environment boots successfully, but you are not happy with the results	Run the `luactivate` command with the name of the previous boot environment and reboot
If the new boot environment does not boot	Boot the fallback boot environment in single-user mode and run the `luactivate` command and reboot
If you cannot boot in single-user mode	Perform 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 145.

Figure 4–11 shows the switch that is made when you reboot to fallback.

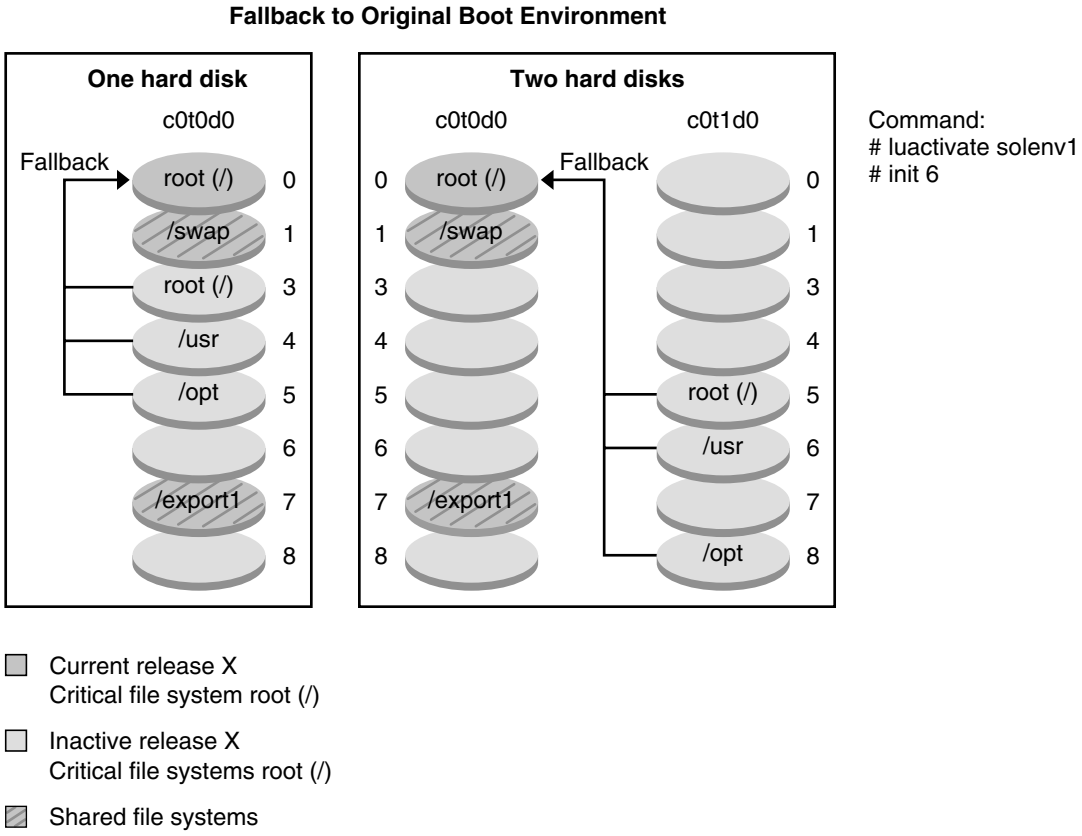


FIGURE 4–11 Fallback to the Original Boot Environment

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 8.

CHAPTER 5

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 about upgrading in "Upgrade" on page 29. This chapter contains the following sections:

Solaris Live Upgrade Requirements

Before you install and use Solaris Live Upgrade, become familiar with these requirements.

Solaris Live Upgrade System Requirements

Solaris Live Upgrade is included in the Solaris 10 software. You need to install the Solaris Live Upgrade packages on your current OS. The release of the Solaris Live Upgrade packages must match the release of the OS you are upgrading to. For example, if your current OS is the Solaris 8 release and you want to upgrade to the Solaris 10 release, you need to install the Solaris Live Upgrade packages from the Solaris 10 release.

Table 5–1 lists releases that are supported by Solaris Live Upgrade.

TABLE 5–1 Supported Solaris Releases

Platform	Your Current Release	Compatible Upgrade Release
SPARC based system	Solaris 7 or Solaris 8 OS	Solaris 8 or 9 OS
SPARC based system	Solaris 7, Solaris 8, Solaris 9 OS	Solaris 10 OS
x86 based system	Solaris 7 OS	Solaris 8 OS
x86 based system	Solaris 7 or Solaris 8 OS	Solaris 9 OS
x86 based system	Solaris 7, Solaris 8, or Solaris 9 OS	Solaris 10 OS

Installing Solaris Live Upgrade

You can install the Solaris Live Upgrade packages by using the 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 10 Operating System DVD, the Solaris 10 Software - 2 CD, or a net installation image.

Note – If you are running the Solaris 7, Solaris 8, or Solaris 9 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 recommended patch cluster for Java 2 runtime environment. The patch cluster is available on http://sunsolve.sun.com.

For instructions about installing the Solaris Live Upgrade software, see "Installing Solaris Live Upgrade" on page 89.

Solaris Live Upgrade Disk Space Requirements

Follow general disk space requirements for an upgrade. See Chapter 2.

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 that 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 about creating separate file systems for sharing, see "Guidelines for Selecting Slices for Shareable File Systems" on page 81.

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). Solaris Live Upgrade does not implement the full functionality of Solaris Volume Manager, but does require the following components of Solaris Volume Manager.

TABLE 5–2 Required Components for Solaris Live Upgrade and RAID-1 Volumes

Requirement	Description	For More Information
You must create at least one state database and at least three state database replicas.	A state database stores information about 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 information about creating a state database, see Chapter 6, "State Database (Overview)," in *Solaris Volume Manager Administration Guide*.
Solaris Live Upgrade supports only a RAID-1 volume (mirror) with single-slice concatenations on the root (/) file system.	A concatenation is 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 RAID-1 volume A RAID—1 volume can be comprised of a maximum of three concatenations.	For guidelines about creating mirrored file systems, see "Guidelines for Selecting Slices for Mirrored File Systems" on page 78.

Managing Packages and Patches With Solaris Live Upgrade

The following sections list packages that are required by Solaris Live Upgrade and provide information about recommended 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 about adding and removing packages with Solaris Live Upgrade, see the man page, `luupgrade`(1M).
- For more information about packaging requirements, see Appendix B.

Required Packages

Check your current OS 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 5–3 Required Packages for Solaris Live Upgrade

Solaris 7 Release	Solaris 8 Release	Solaris 9 Release
SUNWadmap	SUNWadmap	SUNWcar
SUNWadmc	SUNWadmc	SUNWkvm
SUNWjvrt	SUNWj2rt	SUNWcsr
SUNWlibC	SUNWlibC	SUNWcsu
	SUNWbzip	SUNWcsd
		SUNWadmap
		SUNWadmc
		SUNWadmfw
		SUNWlibC

TABLE 5–3 Required Packages for Solaris Live Upgrade *(Continued)*

Solaris 7 Release	Solaris 8 Release	Solaris 9 Release
		SUNWj2rt The SUNWj2rt package is needed when upgrading and using any Solaris installation program, but is not needed strictly by Solaris Live Upgrade during any operation.

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. When you use Solaris Live Upgrade, the only downtime the system incurs is that of a reboot. You can add patches and packages to a new boot environment with the `luupgrade` command or with a Solaris Flash archive.

Type of Installation	Description	For More Information
To add patches to a boot environment.	Create a new boot environment and use the `luupgrade` command with the `-t` option.	`luupgrade`(1M).
To add packages to a boot environment.	Use the `luupgrade` command with the `-p` option.	`luupgrade`(1M).

Type of Installation	Description	For More Information
To 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 copy can be installed on multiple systems.	■ For details about how to create a Solaris Flash archive, see Chapter 3, "Creating Solaris Flash Archives (Tasks)," in *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*. ■ For information about using Solaris Live Upgrade to install a Solaris Flash archive, see "Installing Solaris Flash Archives on a Boot Environment" on page 135

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 about adding and removing packages with Solaris Live Upgrade, see the man page, `luupgrade`(1M).
- For more information about packaging requirements, see Appendix B.

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. 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. To create the same number of files systems that is on your current system, you must specify one `-m` option for each file system to be created.

  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 that is specified by the `-m` option. If you specify the `-m` option twice, you create two file systems. 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 OS. 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 reslicing 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
- 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 197.

General Guidelines When Creating RAID-1 Volumes (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 "RAID Volume Name Requirements and Guidelines for Custom JumpStart and Solaris Live Upgrade" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

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 see 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

These guidelines contain configuration recommendations and examples for a swap slice.

Configuring Swap for the New Boot Environment

You can configure a swap slice in three ways by 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 use 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 OS 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.

Reconfiguring a disk	Examples	For More Information
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.	`format`(1M)
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 10 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 10 release. `/home` is shared between the Solaris 8 and Solaris 10 releases.	For a description of shareable and critical file systems, see "File System Types" on page 52.

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 reinstate 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?	Exclude Options	Include Options
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 119.

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. You can choose from three methods to update 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 addition 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 addition 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 by using the CUI, you type **`yes`** when prompted.
- To force synchronization by 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 OS. 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, despite 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 of or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 10 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 OS, the boot to the Solaris 7 release could fail because the Solaris 10 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 6

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.

Interface Type	Description
Character user interface (CUI)	The CUI does not provide access to all features of Solaris Live Upgrade. The CUI does not run in multibyte locales and 8-bit locales.
Command-line interface (CLI)	The CLI procedures in this document cover the basic uses of the Solaris Live Upgrade commands. See Chapter 10 for a list of commands and also see the appropriate, associated man pages for more options to use with these commands.

Using Solaris Live Upgrade Menus (CUI)

FIGURE 6–1 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. For example, the first black rectangle represents F1 and the second black rectangle represents F2. 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 6–1 Task Map: Using Solaris Live Upgrade

Task	Description	For Instructions
Install Solaris Live Upgrade packages	Install packages on your OS	"Installing Solaris Live Upgrade" on page 89
Start Solaris Live Upgrade	Start the Solaris Live Upgrade main menu	"Starting and Stopping Solaris Live Upgrade (Character Interface)" on page 91
Create a boot environment	Copy and reconfigure file systems to an inactive boot environment	"Creating a New Boot Environment" on page 92

Installing Solaris Live Upgrade

You need to install the Solaris Live Upgrade packages on your current OS. The release of the Solaris Live Upgrade packages must match the release of the OS you are upgrading to. For example, if your current OS is the Solaris 8 release and you want to upgrade to the Solaris 10 release, you need to install the Solaris Live Upgrade packages from the Solaris 10 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 90 or "To Install Solaris Live Upgrade With the Solaris Installation Program" on page 90.

Note – If you are running the Solaris 7, Solaris 8, or Solaris 9 OS, 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 recommended patch cluster for the Java 2 runtime environment recommended to run the Solaris Live Upgrade installer and install the packages. To install the Solaris Live Upgrade packages, use the `pkgadd` command. Or, install for the Java 2 runtime environment the recommended patch cluster. The patch cluster is available on http://sunsolve.sun.com.

▼ To Install Solaris Live Upgrade With the `pkgadd` Command

Steps

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 Installation Program

Steps

1. **Insert the Solaris 10 Operating System DVD or Solaris 10 Software - 2 CD.**

2. **Run the installer for the media you are using.**

   - If you are using the Solaris 10 Operating System DVD, change directories to the installer and run the installer.

     ```
     # cd /cdrom/cdrom0/Solaris_10/Tools/installers
     # ./liveupgrade20
     ```

     The Solaris installation program GUI is displayed.

   - If you are using the Solaris 10 Software - 2 CD, run the installer.

```
% ./installer
```

The Solaris installation program GUI 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 installation program 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`.

Steps

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. **Type:**

   ```
   # /usr/sbin/lu
   ```

   The Solaris Live Upgrade main menu is displayed.

FIGURE 6–2 Solaris Live Upgrade Main Menu

▼ To Stop Solaris Live Upgrade Menus

Step ● **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 52.

▼ To Create a Boot Environment (Character Interface)

Steps

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:   solaris10
   ```

3. **To save your changes, press F3.**

   The configuration menu appears.

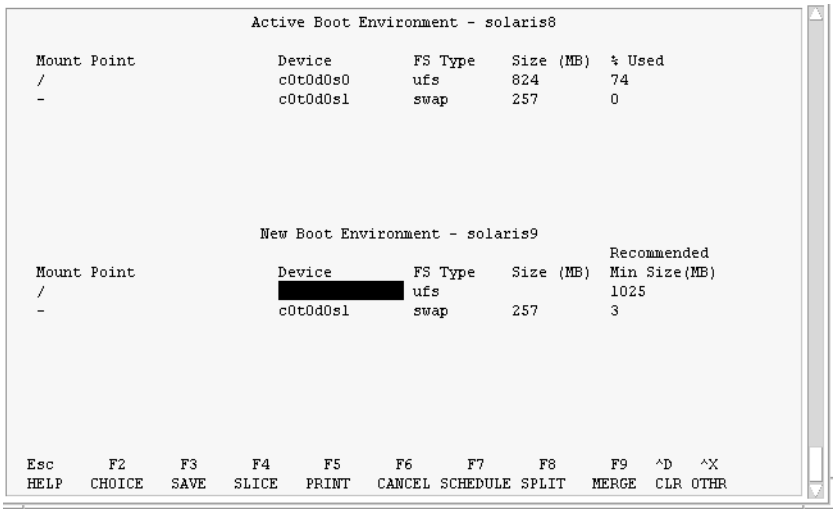


FIGURE 6–3 Solaris Live Upgrade Configuration Menu

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 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 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 52.
- 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 onscreen 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 device 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 Device 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 78.
- 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 Device 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`(1M) command runs, which enables you to create new slices. Follow the screen to create a new slice.

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 OS. 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 about 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 7.

▼ 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 that is 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.

Steps

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. **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 OS 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 applied to 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 113.

`-n` *BE_name*
The name of the boot environment to be created. *BE_name* must be unique on the system.

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 7.

Example 6–1 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
```

▼ To Create a Boot Environment and Merge File Systems (Command-Line Interface)

Note – You can use the `lucreate` command with the `-m` option to specify 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.

Steps

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. **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 applied to 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 113.

`-n` *BE_name*
The name of the boot environment to be created. *BE_name* must be unique on the system.

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 7.

Example 6–2 Creating a Boot Environment and Merging 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
```

▼ 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 OS. 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.

Steps

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. **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 applied to 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 113.

`-n` *BE_name*
The name of the boot environment to be created. *BE_name* must be unique on the system.

Example 6–3 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 7.

▼ To Create a Boot Environment and Reconfiguring 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.

Steps

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. **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 applied to 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 113.

`-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.

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 7.

Example 6–4 Creating a Boot Environment and Reconfiguring 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 106.

▼ 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.

Steps

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 applied to 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 113.

`-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.

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 7.

Example 6–5 Create a Boot Environment and Reconfiguring Swap By 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
```

▼ 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.

Steps

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. **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 applied to 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 113.

`-n` *BE_name*
The name of the boot environment to be created. *BE_name* must be unique.

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 7.

Example 6–6 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.

```
# 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
```

▼ 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.

Steps

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. **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/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 applied to 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 113.

`-n` *BE_name*
The name of the boot environment to be created. *BE_name* must be unique on the system.

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 7.

Example 6–7 Creating a Boot Environment From a Different Source (Command-Line Interface)

In this example, a boot environment is created that is 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
```

▼ 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 that are 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.

Steps

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. **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 applied to 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 113.

`-n` *BE_name*
The name of the boot environment to be created. *BE_name* must be unique on the system.

Example 6–8 Creating an Empty Boot Environment for a Solaris Flash Archive (Command-Line Interface)

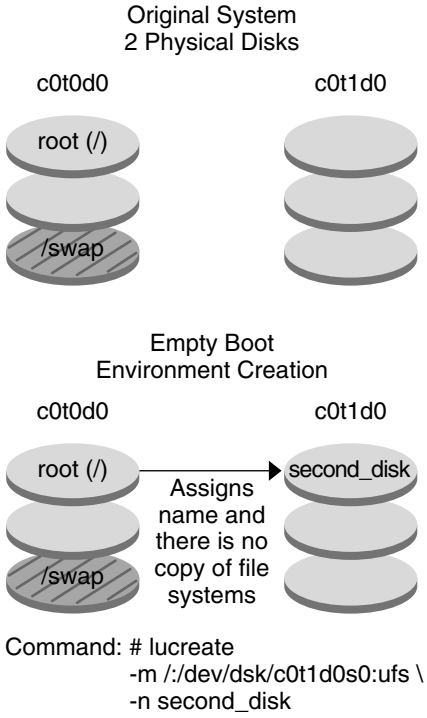
In this example, a boot environment is created but contains no file systems. A description, `mydescription`, is associated with the new boot environment that is named second_disk.

```
# 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 7.

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 175.

The following image shows the creation of 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 about disk about the state of your Solaris Volume Manager configuration.

- For information about creating a state database, see Chapter 6, "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 RAID-1 Volume File Systems" on page 57.
- For in-depth information about complex Solaris Volume Manager configurations that are not allowed when using Solaris Live Upgrade, see Chapter 2, "Storage Management Concepts," in *Solaris Volume Manager Administration Guide*.

Steps

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. **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 applied to 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 content. This keyword enables you to bypass the copying process that copies the content of the source boot environment. Saving the content 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 content is 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 a Solaris Volume Manager volume.
    - If you use the `preserve` keyword when the UFS file system is on a physical slice, the content of the UFS file system is saved on the slice. In the following example of the `-m` option, the `preserve` keyword saves the content 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 are 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 7.

Example 6–9 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.
- 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
```

Example 6–10 Creating a Boot Environment With a Mirror and Not Specifying a Submirror Name (Command Line Interface)

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 7.

Example 6–11 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.
- 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 7.

Example 6–12 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 7.

Example 6–13 Creating a Boot Environment With Two Mirrors (Command Line Interface)

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 7.

▼ 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.

Steps

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. **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 applied to 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 113.

`-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 a 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.

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 7.

Example 6–14 Creating a Boot Environment and Excluding Files (Command Line Interface)

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 6–15 Creating a Boot Environment and Excluding and Including Files (Command Line Interface)

In this example, the new boot environment is named `second_disk`. The source boot environment contains one file system for the OS, 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
```

CHAPTER 7

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 10 and the appropriate man pages, which more fully document CLI options.

Task Map: Upgrading a Boot Environment

TABLE 7–1 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 OS image. ■ Install a Solaris Flash archive on an inactive boot environment.	■ "Upgrading a Boot Environment" on page 124 ■ "Installing Solaris Flash Archives on a Boot Environment" on page 135
Activate an inactive boot environment.	Makes changes effective and switches the inactive boot environment to active .	"Activating a Boot Environment" on page 140
(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 145

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 127.

Guidelines for Upgrading

When you upgrade a boot environment with the latest OS, 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 critical file systems of the inactive boot environment, but shareable file systems are not changed. See "Installing Solaris Flash Archives on a Boot Environment" on page 135.

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 113.

Adding and Removing Packages With Solaris Live Upgrade

Caution – When you are upgrading. 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 about 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 74.
- For information about the command and options, see the man page, `luupgrade`(1M).
- For more information about packaging requirements, see Appendix B.

▼ 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 127.

Steps

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_10/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_10/os_image
     ```

4. **To upgrade, press F3.**
   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 127.

Steps

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. **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 7–1 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_10/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. Use the `luupgrade` command with the `-i` option to install any additional CDs.

Steps

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. **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`	Installs additional CDs. The software 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 140.

Example 7–2 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 10 Software - 1 and the Solaris 10 Software - 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 type information. Omit these options to display the interface.

Insert the Solaris 10 Software - 1 CD and type:

```
# luupgrade -u -n second_disk -s /dev/cdrom/cdrom0/s0
```

Insert the Solaris 10 Software - 2 CD and type:

```
# luupgrade -i -n second_disk -s /dev/cdrom/cdrom0 -O '-nodisplay \
-noconsole'
```

▼ 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 134.
- For a Solaris Flash installation with a profile, see "To Install a Solaris Flash Archive With a Profile (Command-Line Interface)" on page 138.

Steps

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_10`.

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.

   The following table lists the keywords you can use with the `Install_type` keyword values of `upgrade` or `flash_install`.

Keywords for an Initial Archive Creation	Description	Reference
(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	For a description of all the values for this keyword, see "install_type Profile Keyword" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
(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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Keywords for an Initial Archive Creation	Description	Reference
(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)" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
(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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
(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" in *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*.
(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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
(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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

The following table lists the keywords you can use with the `Install_type` keyword value `flash_update`.

Keywords for a Differential Archive Creation	Description	Reference
(Required) `Install_type`	Defines the installation to install a Solaris Flash archive on the system. The value for a differential archive is `flash_update`.	For a description of all the values for this keyword, see "install_type Profile Keyword" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Keywords for a Differential Archive Creation	Description	Reference
(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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
(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.	For more information about this keyword, see "forced_deployment Profile Keyword (Installing Solaris Flash Differential Archives)" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
(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" in *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*.
(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.	For more information about this keyword, see "no_content_check Profile Keyword (Installing Solaris Flash Archives)" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
(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.	For more information about this keyword, see "no_master_check Profile Keyword (Installing Solaris Flash Archives)" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

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 132.

Example 7–3 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
  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 7–4 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 138.

▼ 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.

Step ● **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 that are 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 7–5 Testing a Profile by Using Solaris Live Upgrade

In the following example, the profile is named `flash_profile`. The profile is successfully tested on the inactive boot environment that is 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 10.
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 OS 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 138.

If you added locales to the profile, make sure that you have created a boot environment with additional disk space.

Steps

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. **Create a profile.**

   See "To Create a Profile to be Used by Solaris Live Upgrade" on page 129 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 about creating a profile, see "To Create a Profile to be Used by Solaris Live Upgrade" on page 129.

   The boot environment is ready to be activated.

Example 7–6 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 129.

```
# 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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

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 creating an archive, see Chapter 3, "Creating Solaris Flash Archives (Tasks)," in *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*.

▼ To Install a Solaris Flash Archive on a Boot Environment (Character Interface)

Steps

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_10
Package media: /net/install-svr/export/Solaris_10/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 141.

▼ To Install a Solaris Flash Archive on a Boot Environment (Command-Line Interface)

Steps

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. **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 that are specified with the `-s` option and the `-a` option must be identical.

Example 7–7 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 system versions for the `-s` and `-a` options are both Solaris 10 releases. All files are overwritten on `second_disk` except shareable files.

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_10/OS_image \
-a /net/server/archive/Solaris_10
```

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.

Steps

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. **Create a profile.**

   See "To Create a Profile to be Used by Solaris Live Upgrade" on page 129 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 7–8 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 129.

▼ 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.

Steps

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. **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" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

The boot environment is ready to be activated.

Example 7–9 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` keywords are 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 145.

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 152.
If the boot environment is not the current boot environment, you cannot have mounted the partitions of that boot environment by using the `luumount` or `mount` commands	To view man pages, see `lumount`(1M) or `mount`(1M).
The boot environment that you want to activate cannot be involved in a comparison operation.	For procedures, see"Comparing Boot Environments" on page 157.

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 Reconfiguring Swap (Command-Line Interface)" on page 104.

▼ x86: (Optional) To Update the Boot Diskette Before Activating

If you are not using the Solaris 10 Device Configuration Assistant 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.

Steps

1. **Insert the Solaris 10 Device Configuration Assistant 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. If they are incorrect, 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 about synchronization, see "Synchronizing Files Between Boot Environments" on page 82.

Steps

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_10
   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 of or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 10 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 OS, the boot to the Solaris 7 release could fail because the Solaris 10 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)

Steps

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. **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 7–10 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 about synchronization, see "Synchronizing Files Between Boot Environments" on page 82.

Steps

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. **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 of or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 10 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 OS, the boot to the Solaris 7 release could fail because the Solaris 10 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 7–11 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.

Steps

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. **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 147.

Steps

1. **At the OK prompt, boot the machine to single-user state from the Solaris 10 Operating System DVD, Solaris 10 Software - 1 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 147.
   - 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.

Steps

1. **At the OK prompt, boot the machine to single-user state from the Solaris 10 Operating System DVD, Solaris 10 Software - 1 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.

Steps

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.

Steps 1. **Decide how to boot the system.**

- If you boot from the Solaris 10 Operating System DVD or the Solaris 10 Software - 1 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 Solaris 10 Device Configuration Assistant diskette into the system's diskette drive.

---

x86 only – You can copy the Device Configuration Assistant software to a diskette from the Solaris 10 Operating System for x86 Platforms DVD or Solaris 10 Software for x86 Platforms - 2 CD by using the procedure described in Appendix C.

Follow the directions onscreen 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`):**

   # **`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`.

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 8

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

TABLE 8–1 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 152

TABLE 8–1 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 157 ■ "Displaying the Name of the Active Boot Environment" on page 160 ■ "Viewing the Configuration of a Boot Environment" on page 165
(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 154
(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 159 ■ "Changing the Name of a Boot Environment" on page 161 ■ "Adding or Changing a Description Associated With a Boot Environment Name" on page 162 ■ "Canceling a Scheduled Create, Upgrade, or Copy Job" on page 156

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 from performing live upgrade copy, rename, or upgrade operations.

▼ To Display the Status of All Boot Environments (Character Interface)

Step

- **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)

Steps

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. **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 152 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 156.

▼ To Update a Previously Configured Boot Environment (Character Interface)

Steps

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 about 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 156.

▼ 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.

Steps

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. **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 send an email of the `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 8–1 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
```

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 156.

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)

Steps

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)

Steps

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. **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 152.

The specified boot environment cannot have any partitions that are mounted with `lumount` or `mount`.

▼ To Compare Boot Environments (Character Interface)

Steps

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)

Steps

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. **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 have absolute file names. 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 8–2 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 152 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)

Steps

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)

Steps

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. **Type:**

```
# ludelete BE_name
```

BE_name Specifies the name of the inactive boot environment that is to be deleted

Example 8–3 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 152 to determine a boot environment's status.

▼ To Display the Name of the Active Boot Environment (Character Interface)

Step

- **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)

Step

- **Type:**

```
# /usr/sbin/lucurr
```

Example 8–4 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 152 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)

Steps

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)

Steps

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. **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 about using the `-A` option with `lucreate`	"To Create a Boot Environment for the First Time (Command-Line Interface)" on page 98
For more information about creating the description after the boot environment has been created	`ludesc`(1M)

▼ To Add or Change a Description for a Boot Environment Name With Text

Steps

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. **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 8–5 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 10 test build'
```

▼ To Add or Change a Description for a Boot Environment Name With a File

Steps

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. **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 8–6 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

Steps

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. **Type:**

   ```
   # /usr/sbin/ludesc -A 'BE_description'
   ```

   `-A 'BE_description'` Displays the boot environment name that is associated with the description

Example 8–7 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 10 test build'
 second_disk
```

▼ To Determine a Boot Environment Name From a Description in a File

Steps

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. **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 8–8 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

Steps

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. **Type:**

   ```
   # /usr/sbin/ludesc -n BE_name
   ```

   `-n` *BE_name* — Displays the description that is associated with the boot environment name

Example 8–9 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 10 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)

Steps

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)

Steps

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. **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 9

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 10 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 10 Operating System DVD or Solaris 10 Software - 2 CD.
2. Follow the step for the media you are using.
   - If you are using the Solaris 10 Operating System DVD, change directories to the installer and run the installer.

     ```
     # cd /cdrom/cdrom0/Solaris_10/Tools/Installers
     # ./liveupgrade20
     ```

     The Solaris Web Start installer is displayed.
   - If you are using the Solaris 10 Software - 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 about 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_10 \
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 9–1.
- The activation fails and you can boot back to the original boot environment. See Example 9–2.
- The activation fails and you must boot back to the original boot environment by using media or a net installation image. See Example 9–3.

EXAMPLE 9–1 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`.

```
# /usr/sbin/luactivate first_disk
# init 6
```

EXAMPLE 9–2 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 9–3 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 9–1 shows the current boot environment, which contains three physical disks.

Detaching and Upgrading One Side of a RAID-1 Volume (Mirror)

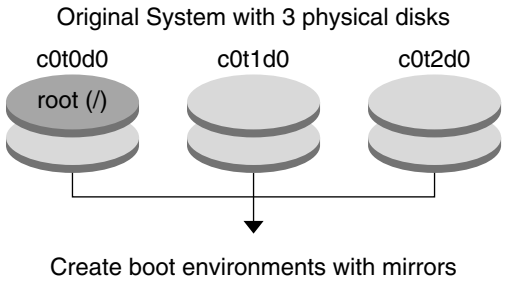


FIGURE 9–1 Detaching and Upgrading One Side of a RAID-1 Volume (Mirror)

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, which 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 system versions for the `-s` and `-a` options are both Solaris 10 releases. All files are overwritten on `third_disk` except shareable files.

```
# luupgrade -f -n third_disk \
-s /net/installmachine/export/Solaris_10/OS_image \
-a /net/server/archive/Solaris_10
```

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 9–2 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)

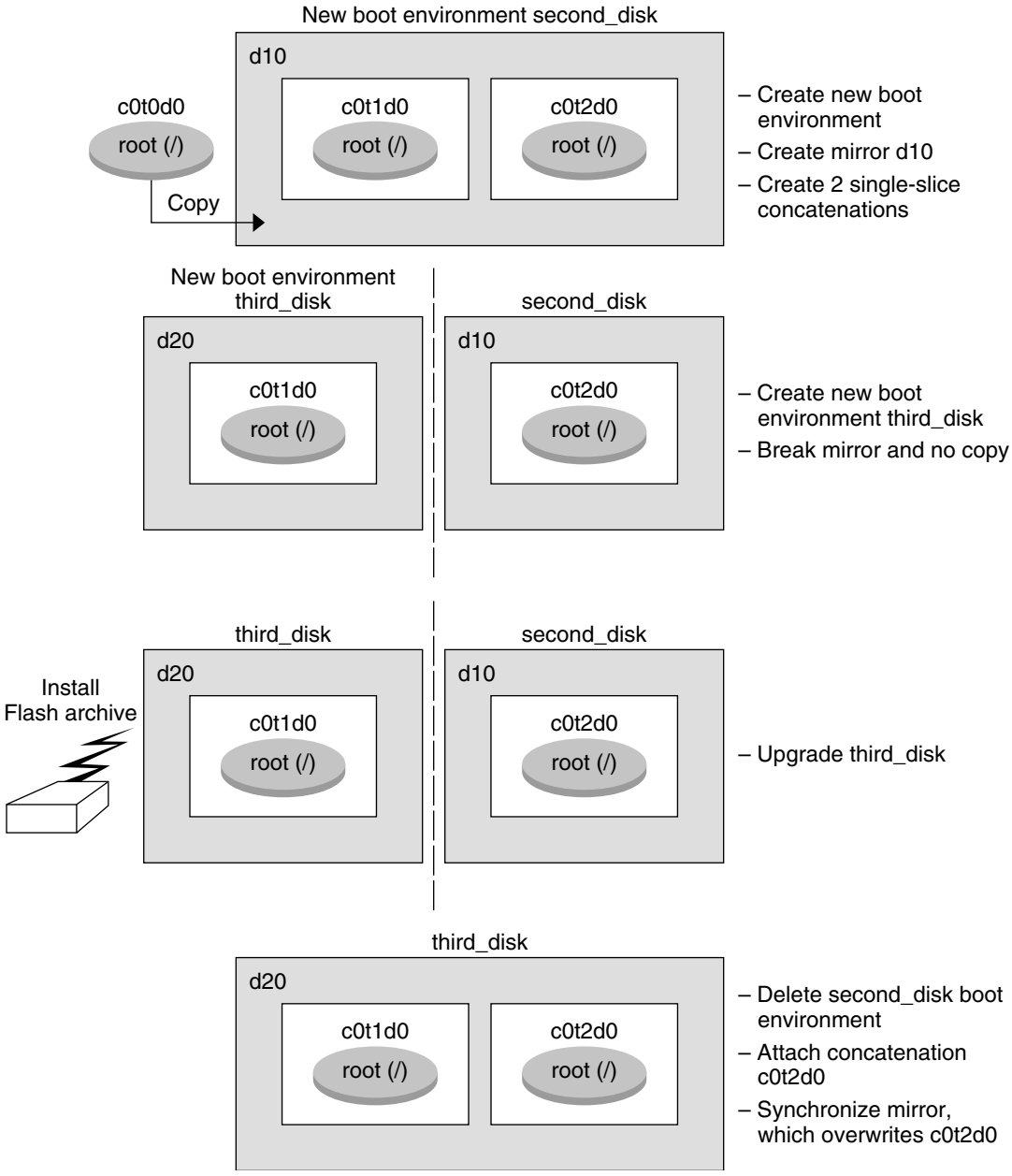


FIGURE 9–2 Detaching and Upgrading One Side of a RAID-1 Volume (Mirror) (continued)

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 cover 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 you use 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 that are 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 9–3 shows the creation of an empty boot environment.

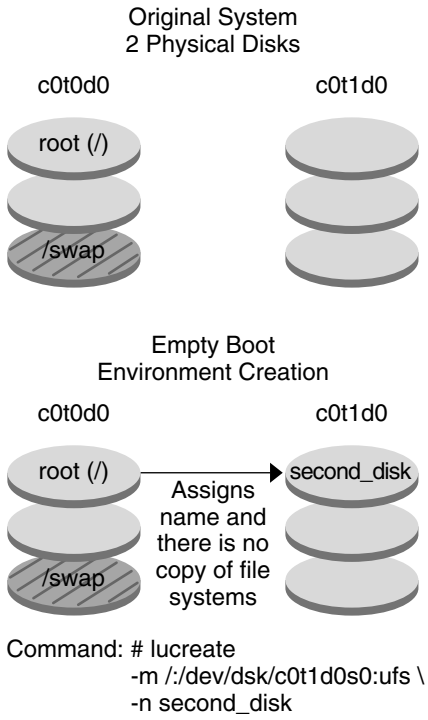


FIGURE 9–3 Creating an Empty Boot Environment

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 previous example. The archive is located on the local system. The operating system versions for the `-s` and `-a` options are both Solaris 10 releases. The archive is named `Solaris_10.flar`.

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_10/OS_image \
-a /net/server/archive/Solaris_10.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 111.
- For step-by-step information about creating a Solaris Flash archive, see Chapter 3, "Creating Solaris Flash Archives (Tasks)," in *Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)*.
- 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 145.

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 10 release. The upgraded boot environment is then activated.

Install Live Upgrade on the Active Boot Environment

1. Insert the Solaris 10 Operating System DVD or Solaris 10 Software - 2 CD.
2. Run the installer for the media you are using.
   - If you are using the Solaris 10 Operating System DVD, change directories to the installer and run the installer.

     ```
     # cd /cdrom/cdrom0/Solaris_10/Tools/Installers
     # ./liveupgrade20
     ```

     The Solaris Web Start installer is displayed.
   - If you are using the Solaris 10 Software - 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 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 instead of sharing the source boot environment's swap slice.

1. Display the character interface:

   ```
   # /usr/sbin/lu
   ```

   The Solaris Live Upgrade Main Menu is displayed.
2. From the main menu, select Create.

   ```
   Name of Current Boot Environment:    c0t4d0s0
   Name of New Boot Environment:   c0t15d0s0
   ```
3. Press F3.

   The Configuration menu is displayed.
4. To select a slice from the configuration menu, press F2.

   The Choices menu is displayed.
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. To select a slice for swap, press F2. The Choices menu is displayed.
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 new boot environment is then upgraded. The new version of the operating system for the upgrade is taken from a network image.

1. From the main menu, select Upgrade.

```
Name of New Boot Environment:   c0t15d0s0
Package Media: /net/ins3-svr/export/Solaris_10/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 169.

CHAPTER 10

Solaris Live Upgrade (Command Reference)

The following list shows commands that you can type at the command line instead of by using the menus. The Solaris Live Upgrade includes man pages for all the listed command-line utilities.

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)

Task	Command
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)
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 OS or install a flash archive on an inactive boot environment.	`luupgrade`(1M)

PART III Appendices

This part provides reference information.

APPENDIX A

Troubleshooting (Tasks)

This chapter contains a list of specific error messages and general problems you might encounter when installing Solaris 10 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.

Note – When you see the phrase "bootable media," this means the Solaris installation program and JumpStart installation method.

Problems With 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.

Description: 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 nonnetworked 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 10 software from the network from an install server. The following are examples of checks you can make.

- If you copied the images of the Solaris 10 Operating System DVD or the Solaris 10 Software CDs to the install server, ensure that you specified the correct platform group for the system when you set it up.
- If you are using DVD or CD media, ensure that the Solaris 10 Operating System DVD or Solaris 10 Software - 1 CD is mounted and accessible on the install server.

`boot: cannot open <`*filename*`> (`*SPARC based systems only*`)`

Cause: This error occurs when you override the location of the `boot -file` by explicitly setting it.

Note – *filename* is a variable for the name of the file affected.

Solution: Follow these instructions:

- 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 10 Operating System DVD or the Solaris 10 Software - 1 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 10 software was installed (either through the Solaris installation program or custom JumpStart), no boot disk was selected. You now must use the Solaris 10 Device Configuration Assistant diskette or edit the BIOS to boot the system.

Solution: Follow these instructions:

- Insert the Solaris 10 Device Configuration Assistant diskette into the system's boot diskette drive (usually drive A). For information about accessing the Solaris 10 Device Configuration Assistant diskette, see Appendix C, Preparing to Boot From the Solaris 10 Device Configuration Assistant or the Network (Tasks).
- If you cannot use the bootable media, go into the BIOS and select the BIOS to boot. See your BIOS documentation for instructions.

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 about 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

Description: If your system has a Toshiba SD-M1401 DVD-ROM with firmware revision 1007, the system cannot boot from the Solaris 10 Operating System DVD.

Solution: Apply patch 111649–03, or later version, to update the Toshiba SD-M1401 DVD-ROM drive's firmware. The patch 111649–03 is available on http://sunsolve.sun.com.

The system hangs or panics when nonmemory PC cards are inserted. (*x86 based systems only*)

Cause: Nonmemory PC cards cannot use the same memory resources that are 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 10 Device Configuration Assistant diskette during the pre-booting phase. (*x86 based systems only*)

Solution: Follow these instructions:

- If you are using old drives, they might be unsupported. Check your hardware manufacturer's documentation.
- Make sure the ribbon and power cables are connected 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, designate one drive as the master by setting jumpers (or as a master operating with a slave), and set the second drive as a slave by setting jumpers.
- If one drive is a hard disk and the second a CD-ROM drive, designate one drive as the slave drive by setting jumpers. You can designate either physical drive as the slave drive.
- If problems persist with two drives on a single controller, attach one drive at a time to verify that each drive works. Designate the drive as master or single master by setting jumpers, and use the drive connector at the end of the IDE ribbon cable to attach the drive. Verify that each drive works, then set the jumpers for 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 10 Device Configuration Assistant diskette during the pre-booting phase.` (*x86 based systems only*)

Solution: Follow these instructions:

- If disks are disabled in the BIOS, use the Solaris 10 Device Configuration Assistant diskette to boot from the hard disk. For information about accessing the Solaris 10 Device Configuration Assistant, see Appendix C, Preparing to Boot From the Solaris 10 Device Configuration Assistant or the Network (Tasks).
- 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" in *Solaris 10 Installation Guide: Network-Based Installations*.

`prom_panic: Could not mount file system` (*SPARC based systems only*)

Cause: This error occurs when you are installing Solaris from a network, but the boot software cannot locate the following:

- Solaris 10 Operating System DVD, either the DVD or a copy of the DVD image on the install server
- Solaris 10 Software - 1 CD image, either the Solaris 10 Software - 1 CD or a copy of the CD image on the install server

Solution: Ensure that the installation software is mounted and shared.

- If you are installing Solaris from the install server's DVD-ROM or CD-ROM drive, ensure that the Solaris 10 Operating System DVD or Solaris 10 Software - 1 CD is inserted in the CD-ROM drive, is mounted, and is shared in the `/etc/dfs/dfstab` file.
- If installing from a copy of the Solaris 10 Operating System DVD image or Solaris 10 Software - 1 CD image on the install server's disk, ensure that the directory path to the copy is shared in the `/etc/dfs/dfstab` file.

`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 installation program first looks in the NIS maps for `bootparams` information. If the program does not find any information, the installer 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 i86pc 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.

Initial Installation of the Solaris Operating System

`Initial installation fails`

Solution: If the Solaris installation fails, you must restart the installation. To restart the installation, boot the system from the Solaris 10 Operating System DVD, the Solaris 10 Software - 1 CD, 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_10/SUNW`*xxxx*`/reloc.cpio: Broken pipe`

Description: This error message is informational and does not affect the installation. The condition occurs when a write on a pipe does not have a reading process.

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 10 Device Configuration Assistant 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 10 Device Configuration Assistant 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 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.

Steps

1. **Boot to the installation media in single-user mode.**

   ```
   # b -s
   ```

2. **Start the `format`(1M) 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 remaps them.

8. **Type:**

   ```
   analyze> quit
   ```

9. **Do you want to specify blocks to remap?**

   - 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 Operating System

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 Software 1 CDROM. (x86 based systems only)`

Cause: You cannot upgrade with Solaris 10 Software - 1 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 Solaris installation from Solaris 10 Operating System DVD or a net installation image, or 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: Follow these instructions:

- If the file system is not a RAID-1 volume, comment out in the `vsftab` file.
- If the file system is a RAID-1 volume, break the mirror and reinstall. For information about unmirroring, see "Removing RAID-1 Volumes (Unmirroring)" in *Solaris Volume Manager Administration Guide*.

`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.

Solution: 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 installation 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 "Upgrading With Disk Space Reallocation" on page 31 for the space problem and see if you can fix it without using auto-layout to reallocate space.

`Problems upgrading RAID–1 volume root (/) file systems`
Solution: If you have problems upgrading when using Solaris Volume Manager RAID-1 volumes that are the root (`/`) file system, see Chapter 25, "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.

Steps

1. **Reboot the system from the Solaris 10 Operating System DVD, the Solaris 10 Software - 1 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 you use 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.

Steps

1. **Create an inactive boot environment. See "Creating a New Boot Environment" on page 92.**
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 7.**

4. **Activate the inactive boot environment. See "Activating a Boot Environment" on page 140.**

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 the 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 10 OS on a system that does not currently include a service or diagnostic 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 10 OS.

If you installed the Solaris 8 2/02 OS on a system with a service partition, 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 OS, you might not be able to recreate the service partition and upgrade to the Solaris 10 OS.

If you want to include a service partition on the disk that contains the Solaris partition, choose one of the following workarounds.

- To install the software from a net installation image or from the Solaris 10 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 diagnostics CD for your system.

     For information about how to create the service partition, see your hardware documentation.
  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 Solaris installation program to install from the Solaris 10 Software - 1 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 diagnostics CD for your system.

     For information about how to create the service partition, see your hardware documentation.
  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 **B**

Additional SVR4 Packaging Requirements (Reference)

This appendix is for system administrators who install or remove packages, especially third-party packages. Following these packaging requirements enables the following:

- Avoids modifying the currently running system so you can upgrade with Solaris Live Upgrade and create and maintain non-global zones and diskless clients
- Prevents a package from being interactive to automate installations when using installation programs such as custom JumpStart

This chapter contains the following sections:

Preventing Modification of the Current OS

Following the requirements in this section keeps the currently running OS unaltered.

Using Absolute Paths

For an installation of an operating system to be successful, packages must recognize and correctly respect alternate root (/) file systems, such as a Solaris Live Upgrade inactive boot environment.

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`.

Using the pkgadd -R Command

Packages being installed by using the pkgadd -R option or being removed using the pkgrm -R option must not alter the currently running system. This feature is used by custom JumpStart, Solaris Live Upgrade, non-global zones and diskless client.

Any procedure scripts that are included in the packages being installed with the pkgadd command -R option or being removed by 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 that is 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 B–1 provides examples of script syntax.

TABLE B–1 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. Nonrelocatable 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 nonrelocatable), 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
```

Guidelines for Writing Scripts

Your package procedure scripts must be independent of the currently running OS to prevent modifying the OS. 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`.

TABLE B–2 Guidelines For Creating Scripts

Guidelines	Affects Solaris Live Upgrade	Affects non-global zones
Scripts must be written in Bourne shell (`/bin/sh`). Bourne shell is the interpreter that is used by the `pkgadd` command to execute the procedure scripts.	X	X
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.	X	X
Scripts are free to use other standard UNIX commands such as `expr`, `cp`, and `ls` and other commands that facilitate shell scripting.	X	X
Any commands that a script invokes must be available in all releases from Solaris 7 or later, since a package must run on all of those releases. Therefore, you cannot use commands that were added or removed after the Solaris 7 release. For example, the `pgrep` command cannot be used, because this command was added in the Solaris 8 release and not available in the Solaris 7 release. To verify that a specific command or option is supported in a Solaris 7 or later release, see the specific version of *Solaris Reference Manual AnswerBook* on http://docs.sun.com.	X	

Maintaining Diskless Client Compatibility

Packages must not execute commands delivered by the package itself. This is to maintain diskless client compatibility and avoids running commands that might require shared libraries that are not installed yet.

Verifying Packages

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

EXAMPLE B–1 Testing a Package

After a package is created, it must be tested by installing it in an alternate root (/) location by using the `-R` *dir_name* option to `pkgadd`. After the package is installed, it must be checked for correctness by using `pkgchk`, as in this example.

```
# pkgadd -d . -R /a SUNWvxvm
# pkgchk -R /a SUNWvxvm
```

No errors should be displayed.

EXAMPLE B–2 Testing a Package on `/export/SUNWvxvm`

If a package exists at `/export/SUNWvxvm`, then you would issue the following command.

```
# pkgchk -d /export SUNWvxvm
```

No errors should be displayed.

Other commands can check the package when you are creating, modifying, and deleting files. The following commands are some examples.

- 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}
```

Preventing User Interaction When Installing or Upgrading

Packages must be added or removed without the user being prompted for information when using the following standard Solaris utilities.

- The custom JumpStart program
- Solaris Live Upgrade
- Solaris installation program
- Solaris Zones

To test a package to ensure that it will install with no user interaction, a new administration file can be set up 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 page `admin`(4) or `pkgadd`(1M).

The following examples show how the `pkgadd` command uses the 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
  ```

EXAMPLE B–3 Installation Administration File

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
```

EXAMPLE B–3 Installation Administration File *(Continued)*

```
runlevel=nocheck
idepend=nocheck
space=ask
setuid=nocheck
confiict=nocheck
action=nocheck
basedir=default
```

For More Information

The following references provide background information about packaging requirements and specific command syntax.

For more specific information about packaging requirements and definitions of terminology	Chapter 6, "Advanced Techniques for Creating Packages," in *Application Packaging Developer's Guide*
For basic information about adding and removing packages and the installation administration file	Chapter 15, "Managing Software (Overview)," in *System Administration Guide: Basic Administration*
For detailed information about specific commands that are referenced in this appendix, see these man pages	`dircmp`(1), `fssnap`(1M), `ps`(1), or `truss`(1) `pkgadd`(1M), `pkgchk`(1M), or `pkgrm`(1M)
For an overview of Solaris Live Upgrade	Chapter 4
For an overview of custom JumpStart	Chapter 3, "Custom JumpStart (Overview)," in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*
For an overview of Solaris Zones	Chapter 16, "Introduction to Solaris Zones," in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*

APPENDIX C

x86: Preparing to Boot From the Solaris 10 Device Configuration Assistant or the Network (Tasks)

This appendix describes the following topics.

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 10 Device Configuration Assistant image is found in the Tools directory of either the Solaris 10 Operating System for x86 Platforms DVD or the Solaris 10 Software for x86 Platforms - 2 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 installation image with PXE. For information about these methods of booting, see "x86: Booting and Installing Over the Network With PXE" on page 209.

▼ 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.

Steps

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 10 Operating System for x86 Platforms DVD or the Solaris 10 Software for x86 Platforms - 2 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_10_x86/s2/Solaris_10/Tools
     ```
   - For CD media, type:

     ```
     # cd /cdrom/sol_10_x86/Solaris_10/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.**

See Also If you are installing the Solaris OS from CD or DVD media, see "Performing an Installation or Upgrade With the Solaris Installation Program" in *Solaris 10 Installation Guide: Basic Installations*.

If you are installing the Solaris OS over the network, see "x86: To Create an x86 Install Server" on page 211.

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?

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 10 Software for x86 Platforms - 2 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 C–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 OS provides several methods for installation or upgrade. Choose the installation method that is most appropriate for your environment.	"Choosing a Solaris Installation Method" on page 21
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 3
(Optional) Preconfigure system information.	You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.	Chapter 4, "Preconfiguring System Configuration Information (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*
Create an install server.	Set up a Solaris server to install the Solaris OS from the network.	"x86: To Create an x86 Install Server" on page 211
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: To Add Systems to Install Over the Network by Using PXE" on page 214

TABLE C–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.	Chapter 12, "Planning for DHCP Service (Tasks)," 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 result from the `add_install_client -d` command.	"Creating DHCP Options and Macros for Solaris Installation Parameters" in *Solaris 10 Installation Guide: Network-Based Installations*
Boot the client.	Instruct the client BIOS to boot from the network.	"x86: To Boot the Client Over the Network by Using PXE" on page 219

▼ x86: To Create 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.

Before You Begin

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 based system to serve x86 platform installation images, see "To Create a SPARC Install Server With SPARC or x86 DVD Media" in *Solaris 10 Installation Guide: Network-Based Installations*.
- You are using the Solaris Software CDs to create the network installation image. For instructions about how to use the Solaris 10 Operating System DVD to create a network installation image, see Chapter 6, "Preparing to Install From the Network With DVD Media (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*.
- 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.

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

- Solaris 7 x86 platform OS - Patch ID 107260-03

You need the following media.

- Solaris 10 Software CDs
- Solaris 10 Languages for x86 Platforms CD

Steps

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 10 Software - 1 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_10/Tools
   ```

   In the previous example, **`cdrom0`** is the path to the drive that contains the Solaris OS 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 10 Software disc images. To determine available disk space, use the `df -kl` command.

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

   ```
   # cd /
   ```

7. **Eject the Solaris 10 Software - 1 CD.**

8. **Insert the Solaris 10 Software - 2 CD in the system's CD-ROM drive.**

9. **Change to the `Tools` directory on the mounted CD:**

   ```
   # cd /cdrom/cdrom0/Solaris_10/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 10 Software - 2 CD.**

13. **Repeat Step 8 through Step 12 for each Solaris 10 Software you want to install.**

14. **Insert the Solaris 10 Languages CD in the system's CD-ROM drive.**

15. **Change to the `Tools` directory on the mounted CD:**

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

16. **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

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

    ```
    # cd /
    ```

18. **If you want to patch the files that are located in the miniroot (`/`*install_dir_path*`/Solaris_10/Tools/Boot`) on the net installation 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 C–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 10 Software CDs
- Solaris 10 Languages for x86 Platforms CD

Insert the Solaris 10 Software for x86 Platforms - 1 CD in the system's CD-ROM drive.

```
# mkdir -p /export/home/cdx86
# cd /cdrom/cdrom0/s2/Solaris_10/Tools
# ./setup_install_server /export/home/cdx86
# cd /
```

Eject the Solaris 10 Software for x86 Platforms - 1 CD. Insert the Solaris 10 Software for x86 Platforms - 2 CD in the system's CD-ROM drive.

```
# cd /cdrom/cdrom0/Solaris_10/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
```

Repeat the previous commands for each Solaris 10 Software for x86 Platforms CD that you want to install.

Insert the Solaris 10 Languages for x86 Platforms CD in the system's CD-ROM drive.

```
# cd /cdrom/cdrom0/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
#
```

See Also For information about how to add client systems to install over the network with PXE, see "x86: To Add Systems to Install Over the Network by Using PXE" on page 214.

If you are not using PXE, and your client system is on a different subnet than your install server, you must create a boot server. For more information, see "Creating a Boot Server on a Subnet With a CD Image" in *Solaris 10 Installation Guide: Network-Based Installations*.

For additional information about the `setup_install_server` and the `add_to_install_server` commands, see `install_scripts`(1M).

▼ x86: To Add 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.

Use the following `add_install_client` procedure for setting up an x86 client to install from the network by using PXE.

Before You Begin Each system that you want to install needs to find the following:

- An install server. For instructions about how to create an install server from CD media, see "x86: To Create an x86 Install Server" on page 211.

- A DHCP server. For instructions about how to set up a DHCP server to support network installations, see "Supporting Solaris Network Installation With the DHCP Service" in *System Administration Guide: IP Services*.
- The `sysidcfg` file if you use a `sysidcfg` file to preconfigure system information. For information about how to create a `sysidcfg` file, see Chapter 4, "Preconfiguring System Configuration Information (Tasks)," in *Solaris 10 Installation Guide: Network-Based Installations*.
- A name server if you use a name service to preconfigure system information. For information about how to preconfigure information with a name service, see "Preconfiguring With the Name Service" in *Solaris 10 Installation Guide: Network-Based Installations*.
- The profile in the JumpStart directory on the profile server if you are using the custom JumpStart installation method. For information about how to set up a custom JumpStart installation, see Chapter 4, "Preparing Custom JumpStart Installations (Tasks)," in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Steps

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 about name services, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.
3. **Change to the `Tools` directory on the Solaris 10 CD image on the install server:**

   ```
   # cd /install_dir_path/Solaris_10/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` option 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` option with the `-e` option. For more information about class-specific installations when using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" in *Solaris 10 Installation Guide: Network-Based Installations*.
`-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 10 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 10 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` 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 when using DHCP, see "Creating DHCP Options and Macros for Solaris Installation Parameters" in *Solaris 10 Installation Guide: Network-Based Installations*.
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 "Platform Names and Groups" on page 33.

The previous command notes the vendor options and macros that you need to add to you DHCP server. See "Creating DHCP Options and Macros for Solaris Installation Parameters" in *Solaris 10 Installation Guide: Network-Based Installations* for instructions about how to define these vendor options and macros in your DHCP server.

Example C–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. 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 10 Software for x86 Platforms - 1 CD image in `/export/home/cdx86`.

For more information about how to use DHCP to set installation parameters for network installations, see "Supporting Solaris Network Installation With the DHCP Service" in *System Administration Guide: IP Services*.

```
x86_install_server# cd /export/boot/Solaris_10/Tools
x86_install_server# ./add_install_client -d -s rosemary:/export/home/cdx86 \
SUNW.i86pc i86pc
```

Example C–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_10/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 C–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_10/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.

See Also For information about how to boot and install the system from the network, see "x86: To Boot the Client Over the Network by Using PXE" on page 219.

For additional information about the `add_install_client` command, see `install_scripts`(1M).

▼ x86: To Boot 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.

Before You Begin This procedure assumes that you have completed the following tasks.

- Set up an install server. For instructions about how to create an install server from CD media, see "x86: To Create an x86 Install Server" on page 211.
- Set up a DHCP server. For instructions about how to set up a DHCP server to support network installations, see "Supporting Solaris Network Installation With the DHCP Service" in *System Administration Guide: IP Services*.
- Gathered or preconfigured the information you need to install. You can perform this task in one or more of the following ways.
  - Gather the information in "Checklist for Installation" in *Solaris 10 Installation Guide: Basic Installations*.
  - Create a `sysidcfg` file if you use a `sysidcfg` file to preconfigure system information. For information about how to create a `sysidcfg` file, see "Preconfiguring With the sysidcfg File" in *Solaris 10 Installation Guide: Network-Based Installations*.
  - Set up a name server if you use a name service to preconfigure system information. For information about how to preconfigure information with a name service, see "Preconfiguring With the Name Service" in *Solaris 10 Installation Guide: Network-Based Installations*.

- Create a profile in the JumpStart directory on the profile server if you are using the custom JumpStart installation method. For information about how to set up a custom JumpStart installation, see Chapter 4, "Preparing Custom JumpStart Installations (Tasks)," in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

This procedure also assumes that your system can boot from the network. If your system cannot boot from the network, you must create a boot diskette to install over the network. See "x86: Copying the Boot Software to a Diskette" on page 207 for information about how to create a boot diskette.

Steps

1. **Turn on the system.**
2. **Type the appropriate keystroke combination to enter the system BIOS.**

   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.
3. **In the system BIOS, instruct the system to boot from the network.**

   See your hardware documentation for information about how to set the boot priority in the BIOS.
4. **Exit the BIOS.**

   The system boots from the network.
5. **When prompted, select an installation type.**

   - **To install with the Solaris interactive installation GUI, type 1 and Enter.**
   - **To perform a custom JumpStart installation, type 2 and Enter.**
   - **To install with the Solaris interactive text installer in a desktop session, type 3 and Enter.**
   - **To install with the Solaris interactive text installer in a console session, type 4 and Enter.**

   The installation program begins. During the installation, answer the system configuration questions, if you are prompted.
6. **After the system boots and installs over the network, instruct the system to boot from the disk drive on subsequent boots.**

See Also For information about how to complete an interactive installation with the Solaris installation GUI, see "To Install or Upgrade With the Solaris Installation Program" in *Solaris 10 Installation Guide: Basic Installations*.

Glossary

3DES ([Triple DES] Triple-Data Encryption Standard). A symmetric-key encryption method that 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 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.

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.

boot environment A collection of mandatory file systems (disk slices and mount points) that are critical to the operation of the Solaris OS. 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.
`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 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 that are used to create digital signatures and public-private key pairs. The CA guarantees that the individual who is 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 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 it is 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 OS 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 OS. 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`, and `/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 software is automatically installed on a system that is 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, that 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 that was 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) An application-layer protocol. 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 that are 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 who 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 10 release.

Entire Solaris Software Group Plus OEM Support — A software group that contains the entire Solaris 10 release, plus additional hardware support for OEMs. This software group is recommended when installing Solaris software on SPARC based servers.

`/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 `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.

global zone In Solaris Zones, the global zone is both the default zone for the system and the zone used for system-wide administrative control. The global zone is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled. Administration of the system infrastructure, such as physical devices, routing, or dynamic reconfiguration (DR), is only possible in the global zone. Appropriately privileged processes running in the global zone can access objects associated with other zones. See also Solaris Zones and non-global zone.

hard link A directory entry that references a file on disk. More than one such directory entry can reference the same physical file.

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.

hashing The process of changing a string of characters into a value or key that represents the original string.

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 OS overwrites the system's disk or disks with the new version of the Solaris OS. If your system is not running the Solaris OS, you must perform an initial installation. If your system is running an upgradable version of the Solaris OS, an initial installation overwrites the disk and does not preserve the OS or local modifications.

install server A server that provides the Solaris 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 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 (192.168.0.0, 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 Part I, "Introducing System Administration: IP Services," in *System Administration Guide: IP Services*.

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 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) 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 section 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 (Message Digest 5) 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-global zone A virtualized operating system environment created within a single instance of the Solaris Operating System. One or more applications can run in a non-global zone without interacting with the rest of the system. Non-global zones are also called zones. See also Solaris Zones and global zone.

nonnetworked 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 software is divided into *software groups*, which are each composed of *clusters* and packages.

panel A container for organizing the contents of a window, a 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 installation program. The patch analyzer performs an analysis on your system to determine which (if any) patches will be removed by upgrading to a Solaris 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 stripe or concatenation is the basic building block for mirrors.
Reduced Network Support Software Group	A software group that contains the minimum code that is required to boot and run a Solaris system with limited network service support. The Reduced Networking Software Group provides a multiuser text-based console and system administration utilities. This software group also enables the system to recognize network interfaces, but does not activate network services.
/ (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 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 that 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 DVD or CD images	The Solaris software that is installed on a system, which you can access on the Solaris DVDs or CDs or an install server's hard disk to which you have copied the Solaris 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 installation	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 software and third-party software.
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 Zones	A software partitioning technology used to virtualize operating system services and provide an isolated and secure environment for running applications. When you create a non-global zone, you produce an application execution environment in which processes are isolated from the all other zones. This isolation prevents processes that are running in a zone from monitoring or affecting processes that are running in any other zones. See also global zone and non-global zone.

standalone A computer that does not require support from any other machine.

state database A database that stores information about 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 called 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 OS merges the new version of the Solaris OS 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 OS.
upgrade option	An option that is presented by the Solaris installation program. The upgrade procedure merges the new version of Solaris with existing files on your disk or disks. 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 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 called 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.
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` 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.

`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.

zone — See non-global zone

Index

E

F

I

J

K

L

M

N

T

U

V

W