



Solaris Live Upgrade 2.0 Guide

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part No: 806-7933-10
October 2001

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, Solstice DiskSuite, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303-4900 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, Solstice DiskSuite, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



040210@7940



Contents

Preface 7

Solaris Live Upgrade Overview 11

Solaris Live Upgrade Introduction 11

Solaris Live Upgrade Process 12

Solaris Live Upgrade Planning 19

Solaris Live Upgrade System Requirements 19

Disk Space Requirements 20

Required Packages 20

To Check For Packages on Your System 21

Checking System Patch Levels 21

Guidelines for Selecting Slices for File Systems 21

Guidelines for Selecting a Slice for the root (/) File System 21

Guidelines for Selecting a Slice for a /swap File System 22

Using Live Upgrade From a Remote System 22

Upgrades of Root Mirrors and Metadevices 22

Using Solaris Live Upgrade to Create a Boot Environment 23

Solaris Live Upgrade Character User Interface or Command-line Interface 23

Using Solaris Live Upgrade Menus 24

Task Map: Installing and Creating Using Solaris Live Upgrade 25

Installing Solaris Live Upgrade 25

▼ To Install Solaris Live Upgrade 25

Starting and Stopping Solaris Live Upgrade (Character Interface) 26

- ▼ To Start Solaris Live Upgrade Menus 26
- ▼ To Stop Solaris Live Upgrade 27
- Creating a New Boot Environment 27
 - ▼ To Create a Boot Environment (Character Interface) 28
 - ▼ To Create a Boot Environment for the First Time (Command-Line Interface) 32
 - ▼ To Create a Boot Environment and Merge File Systems (Command-Line Interface) 33
 - ▼ To Create a Boot Environment and Split File Systems (Command-Line Interface) 34
 - ▼ To Create a Boot Environment and Reconfigure Swap (Command-Line Interface) 36
 - ▼ To Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface) 37
 - ▼ To Create a Boot Environment and Copy a Shareable File System (Command-Line Interface) 38

Upgrading With Solaris Live Upgrade 41

- Task Map: Upgrading a Boot Environment 42
- Upgrading a Boot Environment 42
 - ▼ To Upgrade an Operating System Image on a Boot Environment (Character Interface) 43
 - ▼ To Upgrade an Operating System Image on a Boot Environment (Command-Line Interface) 44
 - ▼ To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface) 44
- Installing Web Start Flash Archives on a Boot Environment 46
 - ▼ To Install Web Start Flash Archives on a Boot Environment (Character Interface) 46
 - ▼ To Install Web Start Flash Archives on a Boot Environment (Command-Line Interface) 48
- Activating a Boot Environment 49
 - ▼ x86: (Optional) Update Boot Diskette Before Activating 49
 - ▼ To Activate a Boot Environment (Character Interface) 50
 - ▼ To Activate a Boot Environment (Command-Line Interface) 51
 - ▼ To Activate a Boot Environment and Synchronize Files (Command-Line Interface) 51
- Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface) 52
 - ▼ SPARC: To Fall Back to the Original Boot Environment 52

- ▼ SPARC: To Fall Back to the Original Boot Environment by Using a CD or Net Image 53
- ▼ x86: To Fall Back With Boot Environments on Different Disks 54
- ▼ x86: To Fall Back With Boot Environments on the Same Disk 55

Maintaining Solaris Live Upgrade Boot Environments 57

- Overview of Solaris Live Upgrade Maintenance 57
- Displaying the Status of All Boot Environments 58
 - ▼ To Display the Status of All Boot Environments (Character Interface) 59
 - ▼ To Display the Status of All Boot Environments (Command-Line Interface) 59
- Updating a Previously Configured Boot Environment 60
 - ▼ To Update a Previously Configured Boot Environment (Character Interface) 60
 - ▼ To Update a Previously Configured Boot Environment (Command-Line Interface) 61
- Canceling a Scheduled Create, Upgrade, or Copy Job 62
 - ▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Character Interface) 62
 - ▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Command-Line Interface) 62
- Comparing Boot Environments 62
 - ▼ To Compare Boot Environments (Character Interface) 63
 - ▼ To Compare Boot Environments (Command-Line Interface) 63
- Deleting an Inactive Boot Environment 64
 - ▼ To Delete an Inactive Boot Environment (Character Interface) 64
 - ▼ To Delete an Inactive Boot Environment (Command-line Interface) 65
- Displaying the Name of the Active Boot Environment 65
 - ▼ To Display the Name of the Active Boot Environment (Character Interface) 65
 - ▼ To Display the Name of the Active Boot Environment (Command-Line Interface) 66
- Changing the Name of a Boot Environment 66
 - ▼ To Change the Name of an Inactive Boot Environment (Character Interface) 67
 - ▼ To Change the Name of an Inactive Boot Environment (Command-Line Interface) 67
- Viewing the Configuration of a Boot Environment 67
 - ▼ To View the Configuration of Each Inactive Boot Environment (Character Interface) 67

▼ To View the Configuration a Boot Environment (Command-Line Interface) 68

Solaris Live Upgrade Reference 69

A Troubleshooting 71

Upgrade Problems 71

Cannot Use the Solaris 8 Device Configuration Assistant *Intel Platform Edition* diskette to Boot the New Boot Environment 71

Creation of New Boot Environment Fails When Reconfiguring Swap 72

System Panics When Upgrading On Veritas VxVm 72

Glossary 75

Index 79

Preface

The *Solaris Live Upgrade 2.0 Guide* describes an upgrade method that enables the production environment to continue running while another boot environment is upgraded and tested. The inactive boot environment can then be switched to become the production boot environment. The result is a significant reduction in downtime because of failed software upgrades.

Note – The Solaris operating environment runs on two types of hardware, or platforms — SPARC™ and IA (Intel Architecture). The Solaris operating environment also runs on both 64-bit and 32-bit address spaces. The information in this document pertains to both platforms and address spaces unless called out in a special chapter, section, note, bullet, figure, table, example, or code example.

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at <http://ww1.fatbrain.com/documentation/sun>.

Accessing Sun Documentation Online

The docs.sun.comSM 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`.

Typographic Conventions

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

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with on-screen computer output	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You must be <i>root</i> 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-2 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	#

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 that substantially reduces the usual service outage that is associated with an operating system upgrade. You can duplicate your current running boot environment, then while the original boot environment continues to run, you can upgrade the duplicate. Or, rather than upgrading, you can install a Web Start Flash archive on a boot environment. The original system configuration remains fully functional and unaffected by the upgrade or installation of a Web Start Flash archive. The duplicate boot environment is then activated to become the active boot environment when the system is rebooted. If a failure occurs, you have a safety net. You can quickly fall back to the original boot environment with a simple reboot, thereby eliminating the service outages that are associated with the normal test and evaluation process.

Solaris Live Upgrade enables you to migrate a boot environment to have different file system types, sizes, and layouts, without affecting the configuration of the installation software. You can also maintain on your system multiple installations of software packages, including Solaris operating system packages.

For more information about installing and creating Web Start Flash archives, see “Web Start Flash Installation Feature Topics” in *Solaris 8 Advanced Installation Guide*.

Some understanding of basic system administration is necessary before using Solaris Live Upgrade. For background information on system administration tasks such as managing file systems, mounting, booting, and managing swap, see the *System Administration Guide, Volume 1*.

Solaris Live Upgrade Process

The following is an overview of the tasks necessary to create an inactive boot environment, upgrade it, and switch it to become the active boot environment.

Creating a Boot Environment Overview

Solaris Live Upgrade distinguishes between two file system types: critical file systems and shareable file systems. Critical file systems are required by the Solaris operating environment and are separate mount points in the `vfstab` of the active and inactive boot environments. Examples are `root (/) /usr, /var` or `/opt`. These file systems are always copied from the source to the inactive boot environment. 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. Shareable file systems are shared by default, but you can specify a destination slice and then the file systems are copied.

Swap is a special case of a shareable file system. Like a shareable file system, all files are shared by default. But, like a critical file system, you can split and merge swap slices. You do this by using the character user interface or at the command line by using `lucreate` with the `-m` option. A limitation to splitting and merging swap slices is that 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 slice contains a swap, `ufs`, or any other file system. A swap slice is not required. For procedures on reconfiguring swap, see the procedure, "To Create a Boot Environment (Character Interface)" Step 9, or "To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)" on page 36.

Creating an inactive boot environment entails copying critical file systems to another slice. First you identify an unused slice where the critical file systems 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. For the procedure on formatting a slice from menus, see the procedure, "To Create a Boot Environment (Character Interface)," Step 6.

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 or you can split file systems from their parent directories by specifying different mount points. For procedures on splitting and merging file systems, see the procedure, “To Create a Boot Environment (Character Interface)” Step 7 or Step 8 or “To Create a Boot Environment and Split File Systems (Command-Line Interface)” on page 34 or “To Create a Boot Environment and Merge File Systems (Command-Line Interface)” on page 33.

Note – When you create file systems for a boot environment, the rules are identical to the rules for creating file systems for the Solaris operating environment. Solaris Live Upgrade cannot prevent you from 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 (/)`.

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 (unless you have designated some 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 newly defined directories. Files are synchronized (in a limited way) and the active boot environment is not changed in any way.

Figure 1–1 shows critical file systems that have been copied to a new boot environment. The `root (/)` file system, as well as any file systems such as `/usr`, `/var`, or `/opt` are copied. File systems such as `/export/home` are shared by the active and inactive boot environments. For procedures on creating a new boot environment, see “Creating a New Boot Environment” on page 27.

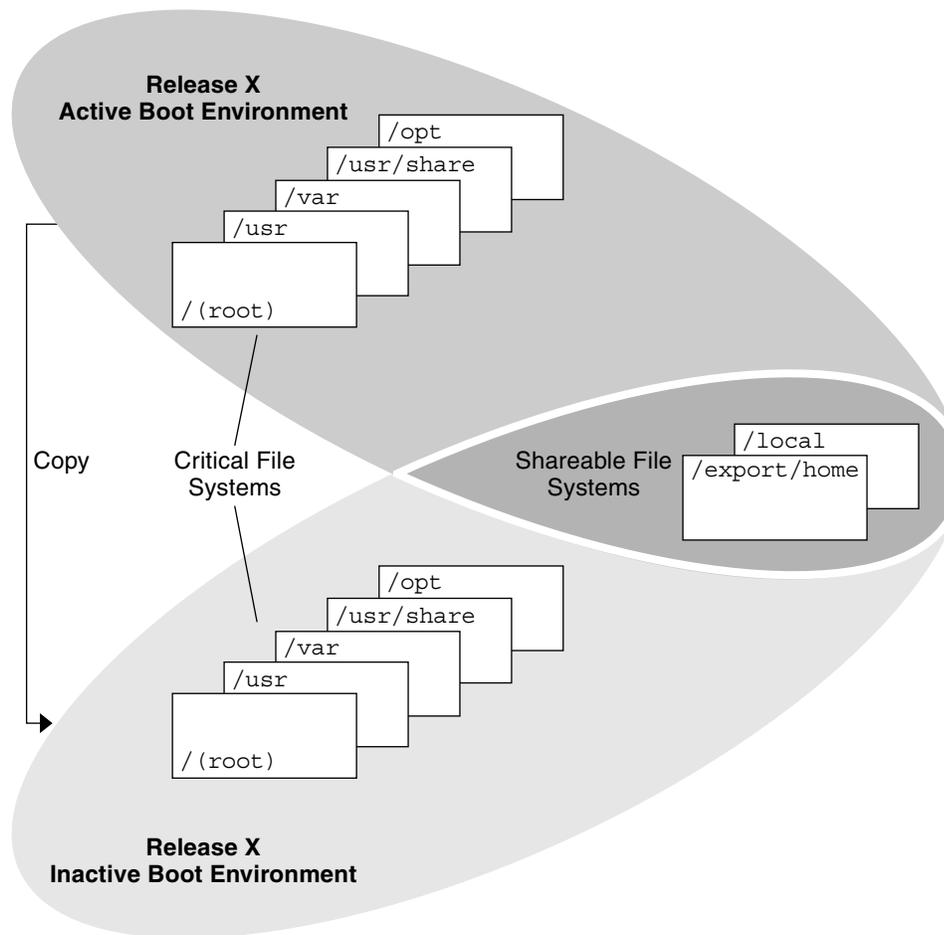


FIGURE 1-1 Creating an Inactive Boot Environment

Upgrading a Boot Environment Overview

After you have created a boot environment, it remains unchanged until you are ready to upgrade it. You can perform an upgrade on the boot environment at any time. The upgrade does not affect any files in the active boot environment. When you are ready, you then activate to the new release. Figure 1-2 shows an upgrade to an inactive boot environment. For procedures on upgrading a boot environment, see Chapter 4.

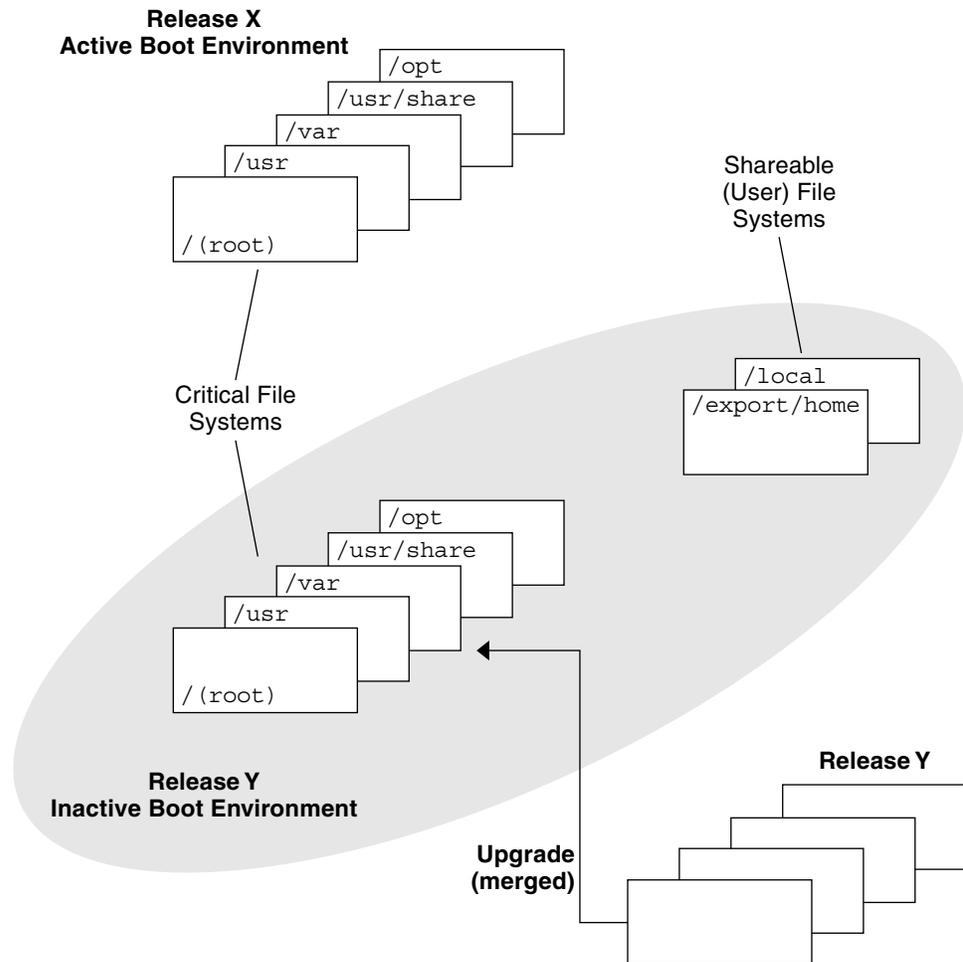


FIGURE 1-2 Upgrading an Inactive Boot Environment

Rather than upgrading, you can install a Web Start Flash archive on a boot environment. The Web Start Flash installation feature enables you to create a single reference installation of the Solaris operating environment on a system that 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. For more information about the Web Start Flash installation feature, see “Web Start Flash Installation Feature Topics” in *Solaris 8 Advanced Installation Guide*.

When you install the Web Start Flash archive on a system, all the files in the archive are copied to that system and a new release is created without affecting the active boot environment. But, unlike an upgrade which merges files, installing a Web Start Flash archive overwrites the files as an initial installation would. Figure 1-3 shows an

installation of a Web Start Flash archive on an inactive boot environment. For procedures on installing a Web Start Flash archive, see “Installing Web Start Flash Archives on a Boot Environment” on page 46.

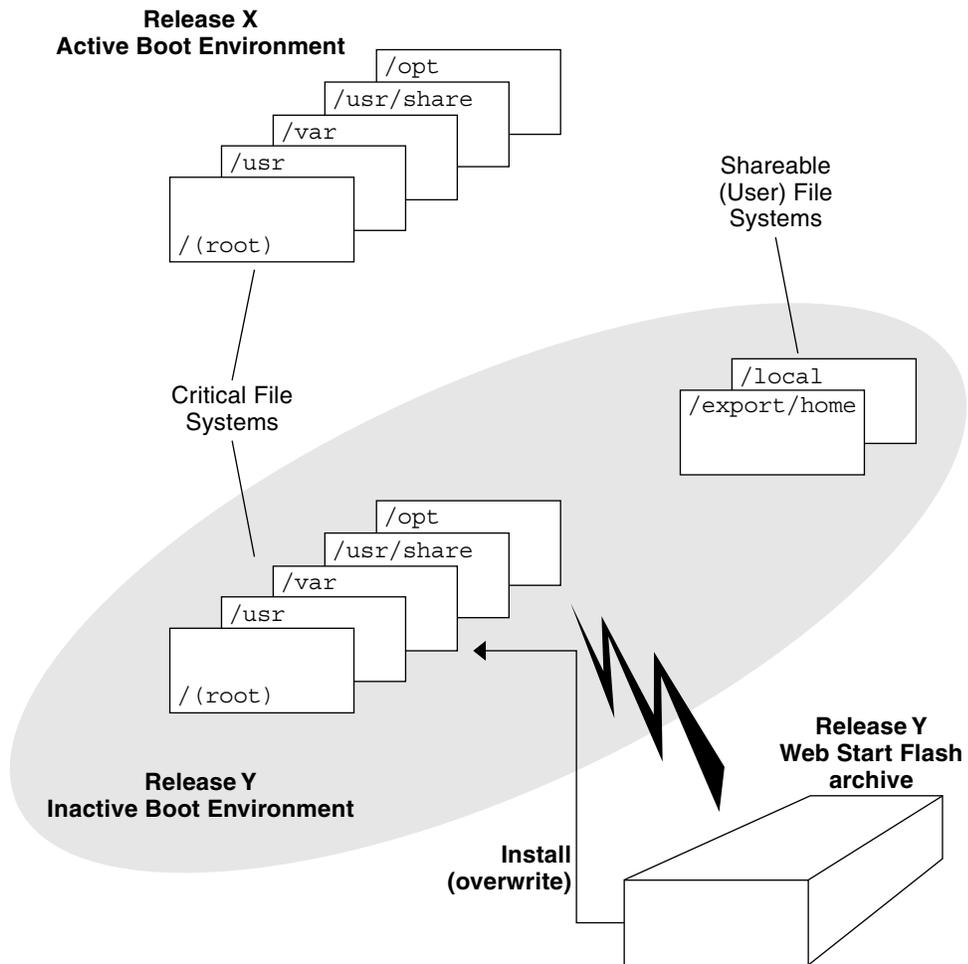


FIGURE 1-3 Installing a Web Start Flash Archive

Activating a Boot Environment Overview

When you are ready to switch and make the boot environment active, you simply activate and reboot. Activating the inactive boot environment modifies it to make it bootable and synchronizes files. When you reboot the system, the configuration that you installed on the inactive boot environment is active. The original boot

environment then becomes an inactive boot environment. Figure 1-4 shows a switch after a reboot from an inactive to an active boot environment. For procedures on activating a boot environment, see “Activating a Boot Environment” on page 49.

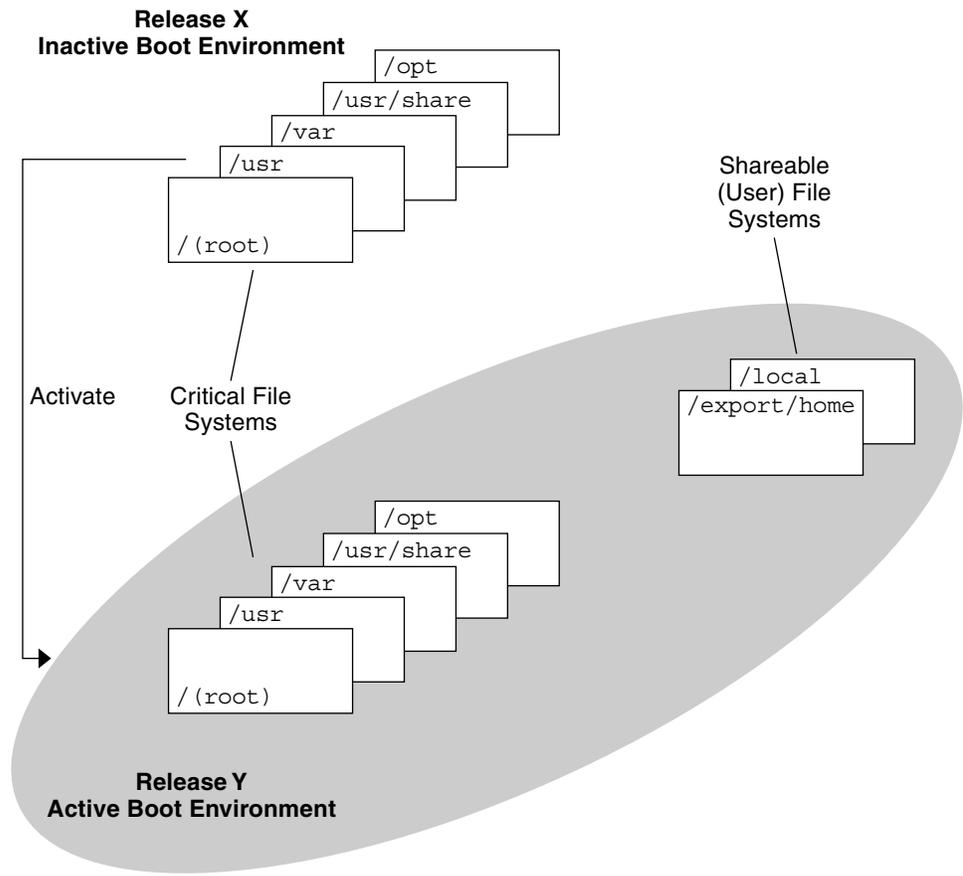


FIGURE 1-4 Activating an Inactive Boot Environment

Fall Back to the Original Boot Environment Overview

If a failure occurs, you can quickly fall back to the original boot environment with an activation and reboot. You need to fall back if the inactive boot environment cannot be booted, or if the environment boots but does not work completely, or you are not satisfied with the results.

The use of fallback rather than a backup and restore of the original takes only the time to reboot the system. The active boot environment is saved and the failure can be analyzed. You can only fall back to the last activated boot environment. To fallback,

you need to find the slice that contains the `root (/)` file system mounted on the last boot environment that was activated. You either run `luactivate` at the command line and then reboot or boot from another media, mount the `root (/)` file system and run `luactivate` on the boot environment you want to fallback to, and reboot. Figure 1-5 shows the switch that is made when you reboot to fallback. For procedures to fallback, see “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 52.

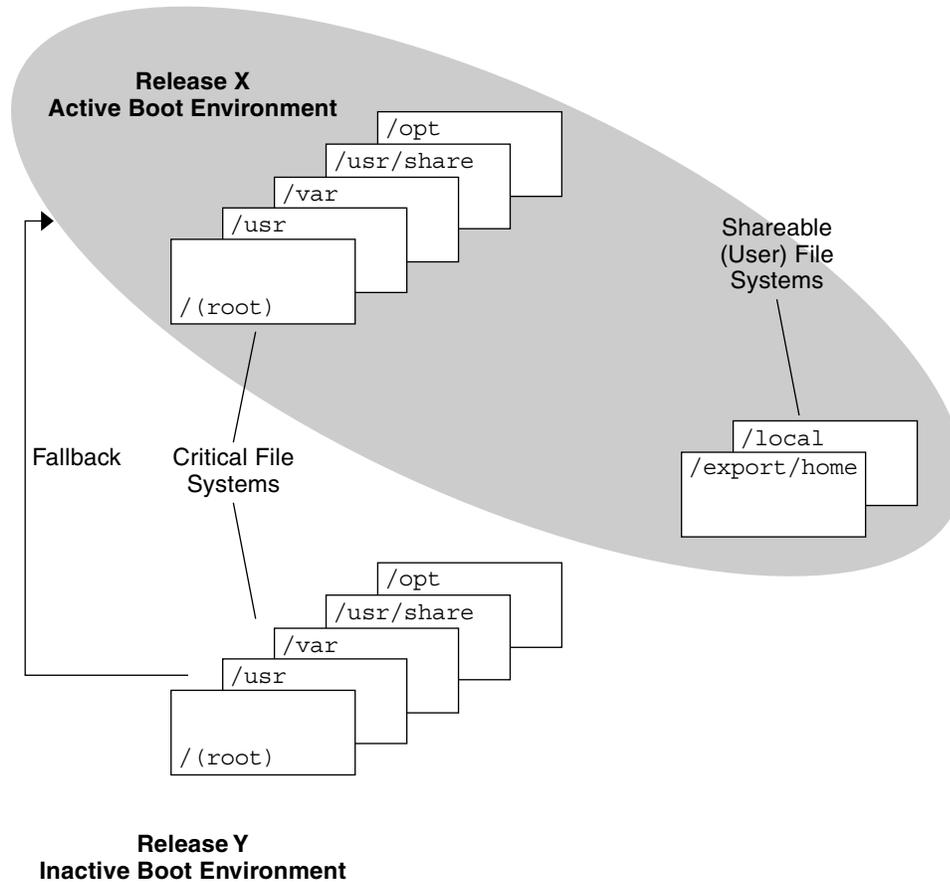


FIGURE 1-5 Fall Back to the Original Boot Environment

Maintaining a Boot Environment Overview

You can also do various maintenance activities such as renaming or deleting a boot environment. For maintenance procedures, see 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 on upgrading in “Checklist for Upgrading” in *Solaris 8 Advanced Installation Guide*. This chapter contains the following sections:

- “Solaris Live Upgrade System Requirements” on page 19
- “Disk Space Requirements” on page 20
- “Checking System Patch Levels” on page 21
- “To Check For Packages on Your System” on page 21
- “Upgrades of Root Mirrors and Metadevices” on page 22

Solaris Live Upgrade System Requirements

- From a SPARC based system, you can upgrade from the Solaris 2.6, Solaris 7, or Solaris 8 operating environment to a Solaris 8 Update operating environment.
- From an IA based system, you can upgrade from the Solaris 7 or 8 operating environment to a Solaris 8 Update operating environment.
- You cannot upgrade to the Solaris 7 operating environment.
- You must upgrade to the same release that contained the Solaris Live Upgrade software. For example, if on your current operating environment you installed Solaris Live Upgrade from the Solaris 8 10/01 release, you must upgrade to the Solaris 8 10/01 release.

Solaris Live Upgrade is included in the Solaris 8 software, but if you want to upgrade from previous releases, you need to install the Solaris Live Upgrade packages on your current operating environment. You can install the Solaris Live Upgrade packages from the installer on the Solaris 8 Software 2 of 2 CD.

For instructions on installing the packages, see “To Install Solaris Live Upgrade” on page 25.

Disk Space Requirements

Typically, each boot environment requires a minimum of 350 to 800 Mbytes of disk space, depending on your system software configuration. Utilities that are used by the Solaris Live Upgrade user interface determine your resource requirements.

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 and you can then abort the process.

You can create a boot environment only on a disk that can 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.

Required Packages

Check your current operating environment for the packages in the following table, which are required to use Solaris Live Upgrade. If packages in the column for your release are missing, use the `pkgadd` command to add these.

TABLE 2-1 Required Packages

Solaris 2.6 Release	Solaris 7 Release	Solaris 8 Release
SUNWadmap	SUNWadmap	SUNWadmap
SUNWadmfw	SUNWadmap	SUNWadmap
SUNWadmcc	SUNWadmcc	SUNWadmcc
SUNWmfrun	SUNWadmcc	SUNWadmcc
SUNWmfrun	SUNWlibC	SUNWlibC
SUNWloc		SUNWbzip

To Check For Packages on Your System

- Type the following to list the packages on your system.

```
% pkginfo
```

Checking System Patch Levels

Solaris Live Upgrade software is designed to be installed and run on multiple versions of the Solaris operating environment. Correct operation of Solaris Live Upgrade requires the latest recommended and security patches for a given OS version. Consult <http://sunsolve.sun.com> for the correct revision level for a patch cluster.

Guidelines for Selecting Slices for File Systems

There are some file system limitations, which are listed below.

When creating file systems for a boot environment, the rules are identical to the rules for creating file systems for the Solaris operating environment. Solaris Live Upgrade cannot prevent you from creating invalid configurations for 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 (/)`.

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.
- Cannot be a Veritas VxVM volume or a Solstice DiskSuite™ metadvice.
- Can be on different physical disks or the same disk as the active root file system.
- If you have a sun4c and sun4m system (not a sun4u UltraSPARC™ system), the `root (/)` file system cannot be greater than 2 Gbytes.

The Choices menu displays most free slices that are available for the creation of an inactive boot environment. Some slices are free, but not available to be shown in the Choices menu, such as a Veritas VxVM volume or a Solstice DiskSuite metadvice.

Guidelines for Selecting a Slice for a /swap File System

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 slice contains a swap, ufs, or any other file system.

Using Live Upgrade From a Remote System

When viewing the character interface remotely, such as over a telnet 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`.

Upgrades of Root Mirrors and Metadevices

If you are using Solaris Live Upgrade on a system that has a Solstice DiskSuite metadvice or Veritas volume, the source boot environment can be a metadvice or volume, but the target boot environment cannot be a metadvice or volume. The inactive boot environment must be a regular slice.

Note – If you have problems upgrading with Veritas VxVM, see “System Panics When Upgrading On Veritas VxVm” on page 72.

Using Solaris Live Upgrade to Create a Boot Environment

This chapter explains how to install Solaris Live Upgrade, use the menus, and to create a boot environment. This chapter contains the following sections:

- “Solaris Live Upgrade Character User Interface or Command-line Interface” on page 23
- “Using Solaris Live Upgrade Menus” on page 24
- “Task Map: Installing and Creating Using Solaris Live Upgrade ” on page 25
- “Installing Solaris Live Upgrade” on page 25
- “Starting and Stopping Solaris Live Upgrade (Character Interface)” on page 26
- “Creating a New Boot Environment” on page 27

Solaris Live Upgrade Character User Interface or Command-line Interface

You can run Solaris Live Upgrade with a Character User Interface (CUI) or at the command line (CLI). The content and sequence of instructions for the CUI and CLI are generally the same. Procedures for both the CUI and CLI are provided in the following sections. See Chapter 6 for a list of commands and also see their appropriate man pages, which more fully document commands.

The CUI does not run in multibyte and 8-bit locales.

Using Solaris Live Upgrade Menus

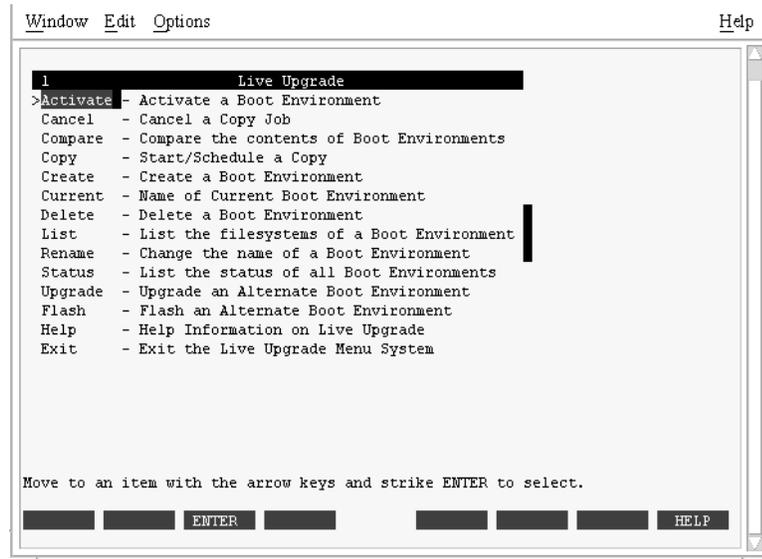


FIGURE 3-1 Solaris Live Upgrade Main Menu

Navigation through the menus of the Solaris Live Upgrade character user interface requires that you use arrow keys and function keys. Use arrow keys to navigate up and down before making a selection or to place the cursor in a field. To perform a task, use the function keys. At the bottom of the menu, you see black rectangles that represent function keys on the keyboard. The first black rectangle represents F1, the second F2, and so on. Rectangles that are active contain a word that represents a task such as Save. The Configuration menu notes the function key number plus the task, rather than a rectangle.

- F3 is always SAVE and completes the task for that menu.
- F6 is always CANCEL and exits the menu without saving changes.
- Other function keys' tasks vary, depending on the menu.

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 when you are asked to press a function key in the following procedures.

Task Map: Installing and Creating Using Solaris Live Upgrade

TABLE 3-1 Task Map: Using Solaris Live Upgrade

Task	Description	For Instructions, Go To
Install Solaris Live Upgrade packages	Install packages on your Solaris 2.6, Solaris 7, or Solaris 8 operating environment	“To Install Solaris Live Upgrade” on page 25
Start Solaris Live Upgrade	Start the Solaris Live Upgrade main menu	“To Start Solaris Live Upgrade Menus” on page 26
Create a boot environment	Copy and reconfigure file systems to an inactive boot environment	“Creating a New Boot Environment” on page 27

Installing Solaris Live Upgrade

If you are upgrading from a Solaris 2.6, Solaris 7, or Solaris 8 operating environment, you need to install the Solaris Live Upgrade packages on your current operating environment.

▼ To Install Solaris Live Upgrade

1. **Insert the Solaris 8 Software 2 of 2 CD.**
2. **Run the installer.**

```
% ./installer
```
3. **From the Select Type of Install panel, click Custom.**
4. **From the Product Selection panel, click the type of installation you want for Solaris Live Upgrade.**

Follow the directions on the Solaris Web Start installer panels to install the software.

Starting and Stopping Solaris Live Upgrade (Character Interface)

This procedure starts and stops the Solaris Live Upgrade menu program.

▼ To Start Solaris Live Upgrade Menus

Note – When viewing the character interface remotely, such as over a tip line, you might need to set the `TERM` environment variable to `VT220`. Also, when using the Common Desktop Environment (CDE), set the value of the `TERM` variable to `dtterm`, rather than `xterm`.

1. **Log in as superuser.**

2. **Type:**

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

The Solaris Live Upgrade main menu is displayed.

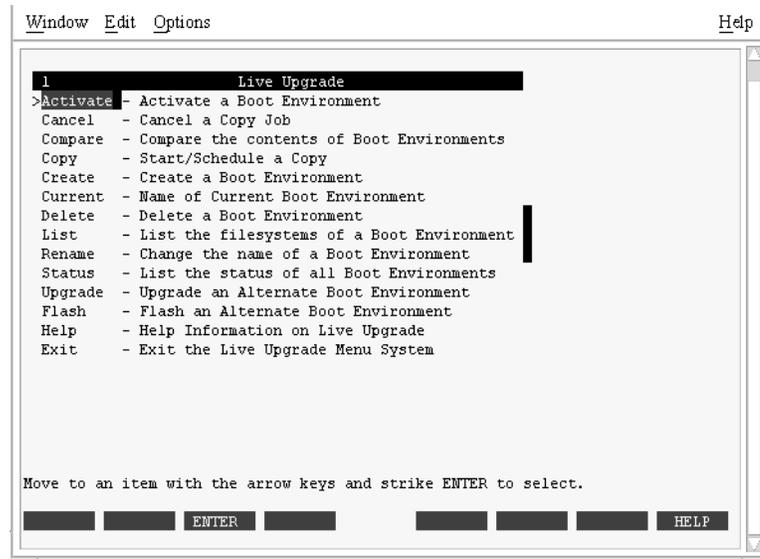


FIGURE 3-2 Solaris Live Upgrade Main Menu

▼ To Stop Solaris Live Upgrade

- Press F6 to exit the main menu.

Creating a New Boot Environment

You can use Solaris Live Upgrade with menus or at the command line (CLI). Procedures are documented for both. These procedures do not exhaust the possibilities for using Solaris Live Upgrade. For more information about commands, see Chapter 6 for commands and the appropriate man pages, which more fully document the CLI.

Creating a boot environment provides a method of copying critical file systems from an active boot environment to an new boot environment. The Create menu, Configuration submenu, and `lucreate` command provide a means to reorganize a disk if necessary, customize file systems, and copy 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. The following procedures for creating enable you to merge, split, and define a new slice for a shareable file system, if you want it copied rather than shared. Swap is a shared file system that can be split and merged.

For an overview of critical and shareable file systems, see “Creating a Boot Environment Overview” on page 12.

▼ To Create a Boot Environment (Character Interface)

1. From the main menu, select Create.

The system displays the Create a Boot Environment submenu.

2. Type the name of the active boot environment (if necessary) and the new boot environment and confirm. You are only required to type the name of the active boot environment the first time you create a boot environment.

The boot environment name can be no longer than 30 characters, contain only alphanumeric characters, and contain no multibyte characters.

Name of Current Boot Environment: **solaris7**
 Name of New Boot Environment: **solaris8**

3. To save your changes, press F3.

The configuration menu appears.

The screenshot shows a terminal window titled "Active Boot Environment - solaris8". It contains two tables. The first table shows the configuration for the active environment (solaris8), and the second table shows the configuration for the new environment (solaris9). At the bottom, there is a row of function key shortcuts.

Active Boot Environment - solaris8				
Mount Point	Device	FS Type	Size (MB)	% Used
/	c0t0d0s0	ufs	824	74
-	c0t0d0s1	swap	257	0

New Boot Environment - solaris9				
Mount Point	Device	FS Type	Size (MB)	Recommended Min Size(MB)
/	[REDACTED]	ufs		1025
-	c0t0d0s1	swap	257	3

Esc	F2	F3	F4	F5	F6	F7	F8	F9	^D	^X
HELP	CHOICE	SAVE	SLICE	PRINT	CANCEL	SCHEDULE	SPLIT	MERGE	CLR	OTHR

FIGURE 3-3 Solaris Live Upgrade Configuration Menu

You see your original boot environment at the top of the screen and the boot environment to be created at the bottom. The area for selecting a critical file system in the Device field is blank until you select a critical file system. Shareable file systems such as `/export` or `swap` are displayed in the Device field and are shared (contain the same mount point) by both the source and target boot environments. In the new boot environment, you can split or merge critical file systems such as `/usr`, `/var`, or `/opt` file systems with the root file system. You can also change the file system type. The swap file system is a special case. 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 “Creating a Boot Environment Overview” on page 12.

- The Device field is the name of a disk device, of the form `/dev/dsk/cnumtnumdnumsnun`
- The FS_Type field can be one of the following:
 - `vxf`s indicates a Veritas file system
 - `sw`ap indicates a swap file system
 - `uf`s indicates a UFS file system

4. (Optional) The following tasks can be done at any time:

- To print the information on screen to an ASCII file, press F5.
- To scroll through the file system list, press Control-X.
You can then switch between the file systems of the active and new boot environment and scroll.
- To exit the Configuration menu at any time, press F6.
 - If you are in the Configuration menu, changes are not saved and file systems are not altered.
 - If you are in a Configuration submenu, you return to the Configuration menu.

5. Select an available slice by pressing F2.

The Choices menu displays available slices on the system for the field where the cursor is placed. The menu displays a Slice field and a file system FS_Type field.

- a. Use the arrow keys to place the cursor in a field to select a slice or file system type.**
- When you place your cursor in the Slice field, all free slices are displayed. For root (`/`), Choices only displays free slices that meet the root (`/`) file system limitations. See “Guidelines for Selecting a Slice for the root (`/`) File System” on page 21.
 - 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 thirty percent for room to upgrade.
- Slices not in bold are too small to support the given file system.

b. Press Return to choose a slice.

The slice appears in the Slice field or the file system type changes in the FS_Type field.

6. (Optional) If available slices do not meet the minimum requirements, to resilic any available disks, press F4.

The Solaris Live Upgrade Slice Configuration menu appears.

The `format` command runs, which enables you to create new slices. Follow the screen to create a new slice. For information on the `format` command, see the `format` man page.

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:

(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` with their parent directory.

Note – When creating file systems for a boot environment, the rules are identical to the rules for creating file systems for the Solaris operating environment. Solaris Live Upgrade cannot prevent you from making invalid configurations for 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.


```
Slice /dev/dsk/c0t4d0s0 will not be swap partition.  
Please confirm? [y, n]: y  
The swap slice no longer exists.
```

12. Decide if you want to create the boot environment now or schedule the creation for later:

- Press F3 to create the new boot environment now.
The configuration is saved and you exit the configuration screen. The file systems are copied, the boot environment is made bootable, and an inactive boot environment is created.
Creating a boot environment might take an hour or more, depending on your system configuration. The Solaris Live Upgrade main menu is then displayed.
- If you want to schedule the creation for a later time, type **y**, then the start time, and an email address as in this example.

```
Do you want to schedule the copy? y  
Enter the time in 'at' format to schedule create: 8:15 PM  
Enter the address to which the copy log should be mailed: someone@anywhere.com
```

You are notified of the completion by email.

For information on time formats, see the `at(1)` man page.

You can schedule only one job at a time.

After the creation is complete, the inactive boot environment is ready to be upgraded.

▼ To Create a Boot Environment for the First Time (Command-Line Interface)

Note – 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.

1. Log in as superuser.
2. To create the new boot environment, type:

```
# lucreate -c BE_name -m mountpoint:device:fs_type \  
-n BE_name
```

<code>-c BE_name</code>	Assigns the name <i>BE_name</i> to the current boot environment. This option is required only when the first boot environment is created. If you run <code>lucreate</code> for the first time and you omit <code>-c</code> , you are prompted to name the current boot environment. If you use the <code>-c</code> option following the first boot environment creation, you receive an error message.
<code>-m mountpoint:device:fs_type</code> [-m...]	Specifies the <code>vfstab</code> information for the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks. <ul style="list-style-type: none"> ■ <i>mountpoint</i> can be any valid mount point or <code>-</code> (hyphen), indicating a swap partition. ■ <i>device</i> field can be one of the following: <ul style="list-style-type: none"> ■ The name of a disk device, of the form <code>/dev/dsk/cnumtnumdnumsnm</code>. ■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent. ■ <i>fs_type</i> field can be one of the following: <ul style="list-style-type: none"> ■ <code>ufs</code>, which indicates a UFS file system. ■ <code>vxfs</code>, which indicates a Veritas file system ■ <code>swap</code>, which indicates a swap file system
<code>-n BE_name</code>	The name of the boot environment to be created. <i>BE_name</i> must be unique on the system.

EXAMPLE 3-1 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. The new boot environment is named `second_disk`. Swap, in the new boot environment `second_disk`, is automatically shared from the source, `first_disk`.

```
# lucreate -c first_disk -m /:/dev/dsk/c0t4d0s0:ufs \
-m /usr:/dev/dsk/c0t4d0s1:ufs -n second_disk
```

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

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

1. Log in as superuser.

2. Type:

```
# lucreate -m mountpoint:device:fs_type \
-m mountpoint:device:fs_type -m mountpoint:merged:fs_type
```

<code>-m mountpoint:device:fs_type</code>	<p>Specifies the <code>vfstab</code> information for the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks.</p> <ul style="list-style-type: none"> ■ <code>mountpoint</code> can be any valid mount point or <code>-</code> (hyphen), indicating a swap. ■ <code>device</code> field can be one of the following: <ul style="list-style-type: none"> ■ The name of a disk device, of the form <code>/dev/dsk/cnumtnumdnumsnm</code>. ■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent. ■ <code>fs_type</code> field can be one of the following: <ul style="list-style-type: none"> ■ <code>vxfst</code>, which indicates a Veritas file system ■ <code>swap</code>, which indicates a swap file system ■ <code>ufs</code>, which indicates a UFS file system
---	--

EXAMPLE 3-2 Creating a Boot Environment and Merge File Systems (Command-Line Interface)

In this example, the `/usr/opt` file system is combined with its parent file system `/usr`.

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
-m /usr/opt:merged:ufs
```

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

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

Note – When creating file systems for a boot environment, the rules are identical to the rules for creating file systems for the Solaris operating environment. Solaris Live Upgrade cannot prevent you from creating invalid configurations for 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 (/)`.

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 to that effect and a symbolic link is created to replace the lost hard link.

1. **Log in as superuser.**

2. **Type:**

```
# lucreate -m mountpoint:device:fs_type -m mountpoint:device:fs_type \
-m mountpoint:device:fs_type -m mountpoint:device:fs_type \
```

`-m mountpoint:device:fs_type -n new_BE`

<code>-m mountpoint:device:fs_type</code>	<p>Specifies the <code>vfstab</code> information for the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks.</p> <ul style="list-style-type: none">■ <code>mountpoint</code> can be any valid mount point or <code>-</code> (hyphen), indicating a swap partition.■ <code>device</code> field can be one of the following:<ul style="list-style-type: none">■ The name of a disk device, of the form <code>/dev/dsk/cnumtnumdnumsnun</code>.■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent.■ <code>fs_type</code> field can be one of the following:<ul style="list-style-type: none">■ <code>ufs</code>, which indicates a UFS file system.■ <code>vxfs</code>, which indicates a Veritas file system■ <code>swap</code>, which indicates a swap file system
<code>-n BE_name</code>	<p>The name of the boot environment to be created. <code>BE_name</code> must be unique on the system.</p>

EXAMPLE 3-3 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` all 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
# lucreate -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).

▼ To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)

Swap slices are shared between boot environments by default. By NOT specifying swap with the `-m` option, your current and inactive boot environment share the same swap slices. If you want to reconfigure the new boot environment's swap, use the `-m` option to add or remove swap slices in the new boot environment.

Note – The swap slice cannot be in use by any boot environment except the current boot environment or if the `-s` option is used, the source boot environment. The boot environment creation fails if the swap slice is being used by any other boot environment whether it is a swap, ufs, or any other file system.

You can create a boot environment with the existing swap slices and then edit the `vfstab` file after the creation.

1. Log in as superuser.

2. Type:

```
# lucreate -m mountpoint:device:fs_type \  
-m -:device:swap -n BE_name
```

<code>-m mountpoint:device:fs_type</code>	<p>Specifies the <code>vfstab</code> information for the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks.</p> <ul style="list-style-type: none">■ <code>mountpoint</code> can be any valid mount point or <code>-</code> (hyphen), indicating a swap slice.■ <code>device</code> field can be one of the following:<ul style="list-style-type: none">■ The name of a disk device, of the form <code>/dev/dsk/cnumtnumdnumsnun</code>■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent■ <code>fs_type</code> field can be one of the following:<ul style="list-style-type: none">■ <code>vxfst</code>, which indicates a Veritas file system■ <code>swap</code>, which indicates a swap file system■ <code>ufs</code>, which indicates a UFS file system
<code>-n BE_name</code>	<p>The name of the boot environment to be created. <code>BE_name</code> must be unique.</p>

The new boot environment is created with swap moved to a different slice or device.

EXAMPLE 3-4 Creating a Boot Environment and Reconfigure Swap (Command-Line Interface)

In this example, the new boot environment uses both `/dev/dsk/c0t0d0s1` and `/dev/dsk/c0t4d0s1` as swap slices:

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m -:/dev/dsk/c0t0d0s1:swap \  
-m -:/dev/dsk/c0t0d0s1: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 Using a List (Command-Line Interface)” on page 37.

▼ To Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)

If you have a long list of swap slices, create a swap list. `lucreate` uses this list for the swap slices in the new boot environment.

Note – The swap slice cannot be in use by any boot environment except the current boot environment or if the `-s` option is used, the source boot environment. The boot environment creation fails if the swap slice is being used by any other boot environment whether the swap slice contains a swap, ufs, or any other file system.

1. Create a list a swap slices to be used in the new boot environment. The location and name of this file is user-defined. In this example, the contents 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 -m mountpoint:device:fs_type \  
-M slice_list -n BE_name
```

<code>-m mountpoint:device:fs_type</code>	<p>Specifies the <code>vfstab</code> information for the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks.</p> <ul style="list-style-type: none"> ■ <code>mountpoint</code> can be any valid mount point or <code>-</code> (hyphen), indicating a swap slice. ■ <code>device</code> field can be one of the following: <ul style="list-style-type: none"> ■ The name of a disk device, of the form <code>/dev/dsk/cnumtnumdnumsnm</code>. ■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent ■ <code>fs_type</code> field can be one of the following: <ul style="list-style-type: none"> ■ <code>vxfst</code>, which indicates a Veritas file system ■ <code>swap</code>, which indicates a swap file system ■ <code>ufs</code>, which indicates a UFS file system
<code>-M slice_list</code>	<p>List of <code>-m</code> options, which are collected in the file <code>slice_list</code>. Specify these arguments in the format that is specified for <code>-m</code>. Comment lines, which begin with a hash mark (<code>#</code>), are ignored. The <code>-M</code> option is useful when you have a long list of file systems for a boot environment. Note that you can combine <code>-m</code> and <code>-M</code> options. For example, you can store swap slices in <code>slice_list</code> and specify <code>root (/)</code> and <code>/usr</code> slices with <code>-m</code>.</p> <p>The <code>-m</code> and <code>-M</code> options support the listing of multiple slices for a given mount point. In processing these slices, <code>lucreate</code> skips any unavailable slices and selects the first available slice.</p>
<code>-n BE_name</code>	<p>The name of the boot environment to be created. <code>BE_name</code> must be unique.</p>

EXAMPLE 3-5 Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)

In this example, swap in the new boot environment is the list of slices that are noted in the `/etc/lu/swapslices` file.

```
# lucreate -m /:/dev/dsk/c02t4d0s0:ufs -m /usr:/dev/dsk/c02t4d0s1:ufs \
-M /etc/lu/swapslices -n second_disk
```

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

▼ 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 done to the shareable file system is available to both boot environments.

1. **Log in as superuser.**
2. **Create the boot environment.**

```
# lucreate -m mountpoint:device:fs_type \  
-m mountpoint:device:fs_type -m mountpoint:device:fs_type -n BE_name
```

<i>mountpoint:device:fs_type</i>	<p>Specifies the <code>vfstab</code> information for the new boot environment. The file systems that are specified can be on the same disk or they can be spread across multiple disks.</p> <ul style="list-style-type: none">■ <i>mountpoint</i> can be any valid mount point or <code>-</code> (hyphen), indicating a swap slice.■ <i>device</i> field can be one of the following:<ul style="list-style-type: none">■ The name of a disk device, of the form <code>/dev/dsk/cnumtnumdnumsnun</code>.■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent.■ <i>fs_type</i> field can be one of the following:<ul style="list-style-type: none">■ <code>vxfs</code>, which indicates a Veritas file system■ <code>swap</code>, which indicates a swap file system■ <code>ufs</code>, which indicates a UFS file system
<i>-n BE_name</i>	<p>The name of the boot environment to be created. <i>BE_name</i> must be unique.</p>

EXAMPLE 3-6 Creating a Boot Environment and Copying a Shareable File System (Command-Line Interface)

In this example, a boot environment is created and the `/home` file system is copied to the target boot environment.

```
# lucreate -m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \  
-m /home:/dev/dsk/c0t4d0s4:ufs -n second_disk
```

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

Upgrading With Solaris Live Upgrade

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:

- “Task Map: Upgrading a Boot Environment ” on page 42
- “Upgrading a Boot Environment” on page 42
- “Installing Web Start Flash Archives on a Boot Environment” on page 46
- “Activating a Boot Environment ” on page 49
- “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 52

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 regarding commands, see Chapter 6 for commands and the appropriate man pages, which more fully document the CLI.

Task Map: Upgrading a Boot Environment

TABLE 4-1 Task Map: Upgrading With Solaris Live Upgrade

Task	Description	For Instructions, Go To
Either upgrade a boot environment or install a Web Start Flash archive	<ul style="list-style-type: none">■ Upgrade the inactive boot environment with an operating environment image■ Install a Web Start Flash archive on an inactive boot environment	<ul style="list-style-type: none">■ “Upgrading a Boot Environment” on page 42■ “Installing Web Start Flash Archives on a Boot Environment” on page 46
Activate an inactive boot environment	Makes changes effective and switches the inactive boot environment to active	“Activating a Boot Environment” on page 49
(optional) Switch back in case of a failure 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 52

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:

- NFS server
- Local file
- Local tape
- Local device, including 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 44. For the character user interface, you must use a combined installation image.

When you upgrade a boot environment with the latest operating environment, you do not affect the active boot environment. The new files merge with the inactive boot environment critical file systems, but shareable file systems are not changed.

Rather than upgrading, if you have created a Web Start Flash archive, you could install the archive onto an inactive boot environment. See “Installing Web Start Flash Archives on a Boot Environment” on page 46.

▼ To Upgrade an Operating System Image on a Boot Environment (Character Interface)

For the following procedure, you must use a combined installation image or one CD. 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 44.

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 or CD	Specify the local device and the path to the installation image.

- If you have one CD, type the path to that CD as in this example:

```
Package Media: /cdrom/solaris8/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/solaris8/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)

For the following procedure, you must use a combined installation image or just one CD. 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 44.

1. **Log in as superuser.**
2. **Indicate the boot environment to upgrade and the path to the installation software by typing:**

```
# luupgrade -u -n BE_name -s os_image_path
```

-u	Indicates the upgrade OS function
-n <i>BE_name</i>	Specifies the name of the boot environment that is to be upgraded
-s <i>os_image_path</i>	Specifies the path name of a directory that contains an operating system image

EXAMPLE 4-1 Upgrading an Operating System Image on a Boot Environment (Command-Line Interface)

In this example, the `second_disk` boot environment is upgraded by using a path from the network to a combined installation image.

```
# luupgrade -u -n second_disk \  
-s /net/installmachine/export/solarisX/OS_image
```

▼ To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)

If the operating system image resides on two or more CDs, use this upgrade procedure.

1. **Log in as superuser.**
2. **Indicate the boot environment to upgrade and the path to the installation software by typing:**

```
# luupgrade -u -n BE_name -s os_image_path
```

-u	Indicates to install an OS image
----	----------------------------------

<code>-n BE_name</code>	Specifies the name of the boot environment that is to be upgraded
<code>-s os_image_path</code>	Specifies the path name of a directory containing an operating system image

EXAMPLE 4-2 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 the first CD.

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

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 example, but the `-u` option is replaced by the `-i` option. Also, choose to run the installer on the second CD with menus or in text mode.

- 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 in text mode.

```
# luupgrade -i -n BE_name -s os_image_path -O -nodisplay
```

<code>-i</code>	Looks for an installation program on the specified medium and runs that program. The installer program is specified with <code>-s</code> .
<code>-n BE_name</code>	Specifies the name of the boot environment that is to be upgraded
<code>-s os_image_path</code>	Specifies the path name of a directory that contains an operating system image
<code>-O -nodisplay</code>	(Optional)Runs the installer on the second CD in text mode

EXAMPLE 4-3 Upgrading an Operating System Image From Multiple CDs
(Command-Line Interface)

In this example, the upgrade on `second_disk` boot environment is completed with the second CD and the installer on the second CD is run with menus.

```
# luupgrade -i -n second_disk -s /dev/cdrom/cdrom0
```

If you have more CDs, repeat this step.

The boot environment is ready to be activated. See “Activating a Boot Environment” on page 49.

Installing Web Start Flash Archives on a Boot Environment

This section provides the procedure for using Solaris Live Upgrade to install Web Start Flash archives that are stored on the following:

- HTTP server
- NFS server
- Local file
- Local tape
- Local device, including CD

Installing a Web Start Flash archive overwrites all files on the new boot environment, except for shared files.

To use the Web Start Flash installation feature, you need to have installed the master system and created the Web Start Flash archive. For more information about Web Start Flash, see “Web Start Flash Installation Feature Topics” in *Solaris 8 Advanced Installation Guide*.

▼ To Install Web Start Flash Archives on a Boot Environment (Character Interface)

1. **From the Solaris Live Upgrade main menu, select Flash.**

The Flash an Inactive Boot Environment menu is displayed.

2. **Type the name of the boot environment where you want to install the Web Start Flash archive and the location of the installation media:**

```
Name of Boot Environment: solaris_8
Package media: /net/install-svr/export/s8/latest
```

3. **Press F1 to add an archive.**

An Archive Selection submenu displays an empty list, the name of a single archive, or a list of a number of archives. If more than one archive is listed, you have layered archives. For an explanation of layered archives, see “Web Start Flash Overview and Planning” in *Solaris 8 Advanced Installation Guide*.

You can add and remove archives from the list. The following example shows an empty list.

```
Location                - Retrieval Method
<No Archives added>    - Select ADD to add archives
```

- To install the list with no changes, proceed to Step 4.

- If the Web Start Flash archives list is empty or contains an archive that you do not want to install, proceed to Step a.
- a. **You can add or remove archives.**
 - Press F1 to add an archive to the list. Proceed to Step b.

Note – If you have a list of more than one archive, you have layered archives. For more information, see “Web Start Flash Overview and Planning” in *Solaris 8 Advanced Installation Guide*.

A Select Retrieval Method submenu is displayed.

```
HTTP
NFS
Local File
Local Tape
Local Device
```

- Press F2 to remove an archive from the list. Proceed to Step e.
- b. **On the Select Retrieval Method menu, select the location of the Web Start Flash archive.**

Media Selected	Prompt
HTTP	Specify the URL and proxy information that is needed to access the Web Start Flash archive.
NFS	Specify the path to the network file system where the Web Start Flash archive is located. You can also specify the archive file name.
Local file	Specify the path to the local file system where the Web Start Flash archive is located.
Local tape	Specify the local tape device and the position on the tape where the Web Start Flash archive is located.
Local device	Specify the local device, the path to the Web Start Flash archive, and the type of file system on which the Web Start 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. When the list contains the archives you want to install, press F6 to exit.

4. Press F3 to install the archive or layered archives.

The Web Start Flash archive or layered archives are 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 “Activating a Boot Environment” on page 49.

▼ To Install Web Start Flash Archives on a Boot Environment (Command-Line Interface)

1. Log in as superuser.

2. Type:

```
# luupgrade -f -n BE_name -s os_image_path -J "profile"
```

You must specify one of these options: `-a`, `-j`, or `-J`.

<code>-f</code>	Indicates to upgrade an operating system from a flash archive.
<code>-n BE_name</code>	Specifies the name of the boot environment that is to be upgraded.
<code>-s os_image_path</code>	Specifies the path name of a directory that contains an operating system image. This can be a directory on an installation medium, such as a CD-ROM, or can be an NFS or UFS directory.
<code>-J "profile"</code>	Entry from a JumpStart profile that is configured for a flash installation. See <code>pfinstall(1M)</code> and “Custom JumpStart Installation Topics” in <i>Solaris 8 Advanced Installation Guide</i> for information on the JumpStart software. You must specify one of <code>-a</code> , <code>-j</code> , or <code>-J</code> .
<code>-j profile_path</code>	Path to a JumpStart profile that is configured for a flash installation. See <code>pfinstall(1M)</code> and “Custom JumpStart Installation Topics” in <i>Solaris 8 Advanced Installation Guide</i> for information on the JumpStart software. You must specify one of <code>-a</code> , <code>-j</code> , or <code>-J</code> .
<code>-a archive</code>	Path to the flash archive when the archive is available on the local file system. You must specify one of <code>-a</code> , <code>-j</code> , or <code>-J</code> .

EXAMPLE 4-4 Installing Web Start Flash Archives on a Boot Environment
(Command-Line Interface)

In this example, a flash archive is installed on the `second_disk` boot environment. The `-J` option is used to retrieve the archives. 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 “SPARC: To Fall Back to the Original Boot Environment” on page 52.

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 58.
- If the boot environment is not the current boot environment, you cannot have mounted the partitions of that boot environment by using `lumount (1M)` or `mount(1M)`.
- The boot environment you want to activate cannot be involved in a comparison operation. See “Comparing Boot Environments” on page 62.

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 “Creating a New Boot Environment,” Step 9 or “To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)” on page 36.

▼ x86: (Optional) Update Boot Diskette Before Activating

If you’re not using the Solaris 8 Device Configuration Assistant *Intel Platform Edition* diskette to boot the system, you can skip this procedure. If you use the Device Configuration Assistant to boot, you need to update the boot diskette. This procedure matches a boot diskette to your release by overwriting the existing diskette or writing to a new one.

1. Insert the diskette. This is either the existing diskette that is overwritten or a new diskette.
2. Update the boot diskette with the latest image for this release.
3. Eject the boot diskette.
4. Type:

```
volcheck
```
5. Copy the new boot environment's `boot/solaris/bootenv.rc` file to the diskette.

```
cp /a/boot/solaris/bootenv.rc /floppy/floppy0/solaris/bootenv.rc
```
6. Check the input-device and output-device on the diskette to make sure they are correct. If not then update them.
You are ready to activate the new boot environment.

▼ To Activate a Boot Environment (Character Interface)

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_8`
Do you want to force a Live Upgrade sync operations: `no`
3. You can either continue or synchronize files.
 - Press Return to continue.
No file synchronization is done.
 - If it has been some time since you created the inactive boot environment, you might want to synchronize files. To synchronize files, type:
Do you want to force a Live Upgrade sync operations: `yes`
4. Press **F3** to begin the activation process.
5. Press **Return** to continue.
The new boot environment is activated at the next reboot.
6. To activate the inactive boot environment, reboot:

```
# init 6
```

▼ To Activate a Boot Environment (Command-Line Interface)

1. Log in as superuser.
2. (Optional) To check on which boot environment is activated on the next reboot, type:

```
# /usr/sbin/luactivate
```

3. To activate the boot environment, type:

```
# /usr/sbin/luactivate BE_name
```

<i>BE_name</i>	Specifies the name of the boot environment that is to be activated.
----------------	---

EXAMPLE 4-5 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
```

4. Reboot.

```
# init 6
```

▼ To Activate a Boot Environment and Synchronize Files (Command-Line Interface)

The first time you boot from a newly created boot environment, Live Upgrade software synchronizes this boot environment with the boot environment that was last active. (This is not necessarily the boot environment that was the source for the newly created boot environment.) It does not perform this synchronization after this initial boot, unless you use the `-s` option. Use this option with great caution, because you might not be aware or in control of changes that might have occurred in the last active boot environment.

1. Log in as superuser.
2. (Optional) To check on which boot environment is activated on the next reboot, type:

```
# /usr/sbin/luactivate
```

3. To activate the boot environment, type:

```
# /usr/sbin/luactivate -s BE_name
```

<i>BE_name</i>	Specifies the name of the boot environment that is to be activated.
<code>-s</code>	Forced synchronization.

EXAMPLE 4-6 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
```

4. Reboot.

```
# 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 CD-ROM or a net image. For IA based systems, choose the procedure that is based on where the root (/) file systems are mounted, on the same or different physical disks.

▼ SPARC: To Fall Back to the Original Boot Environment

If you experience a failure while booting the new boot environment, use this procedure to fall back to the original boot environment.

1. Log in as superuser.

2. Type:

```
# /sbin/luactivate
```

- If this command fails to display a prompt, proceed to “SPARC: To Fall Back to the Original Boot Environment by Using a CD or Net Image” on page 53.

- 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.

▼ SPARC: To Fall Back to the Original Boot Environment by Using a CD or Net Image

If you were unsuccessful at using `luactivate` in the previous procedure, use this procedure to boot from a CD or a net image. You need to mount the `root (/)` slice from the last active boot environment. Then run the `luactivate` command, which makes the switch. When you reboot, the last active boot environment is up and running again.

1. At the OK prompt, boot the machine to single-user state from the installation CD, the network, or a local disk:

```
OK boot cdrom -s
```

or

```
OK boot net -s
```

or

```
OK boot disk -s
```

disk The name of the disk and the slice where a copy of the operating system resides, for example `disk1:A`.

2. 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

3. Mount the active boot environment root slice to some directory (such as `/mnt/sbin`):

```
# mount logical_device_name /mnt/sbin
```

logical_device_name Any device, including a Veritas VxVM volume or Solstice DiskSuite metadvice

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/sbin

```
# umount logical_device_name /mnt/sbin
```

logical_device_name Any device, including a Veritas VxVM volume or Solstice DiskSuite metadvice

6. Reboot.

```
# init 6
```

The previous working boot environment becomes the active boot environment.

▼ x86: To Fall Back With Boot Environments on Different Disks

If you experience a failure while booting the new boot environment and the root (/) file systems for the boot environments are on different physical disks, use the following procedure to fall back to the original boot environment.

1. Reboot the machine and enter the appropriate BIOS menus.

- If your boot devices are SCSI, refer to documentation on your SCSI controller on how to enter the SCSI BIOS.
- If the boot devices are maintained by the system BIOS, refer to system BIOS documentation on how to enter the system BIOS.

2. Follow the appropriate BIOS documentation to change the boot device back to the original boot environment's boot device.

3. Save the BIOS changes.

4. Exit BIOS to begin the boot process.

5. At the OK prompt, boot the machine to single-user state.

```
OK b -s
```

6. Type:

```
# /sbin/luactivate
```

7. Reboot.

```
# init 6
```

▼ x86: To Fall Back With Boot Environments on the Same Disk

If you experience a failure while booting and the `root (/)` file systems are on the same physical disk, use the following procedure to fall back to the original boot environment. You need to mount the `root (/)` slice from the last active boot environment. Then run the `luactivate` command, which makes the switch. When you reboot, the last active boot environment is up and running again.

1. Decide how to boot the system:

- Access the Solaris Device Configuration Assistant from the Solaris 8 Installation CD, Solaris 8 Software 1 of 2 CD, or from a net image of the CD. Your system's BIOS must support booting from a CD.
- Insert the Solaris 8 Device Configuration Assistant *Intel Platform Edition* diskette. The boot diskette software can be downloaded and copied to a diskette from the Solaris Developer Connection at: soldc.sun.com/support/dirvers/dca_diskettes.
- Access the Solaris Device Configuration Assistant from the network using Pre-boot eXecution Environment (PXE). PXE enables you to boot a system directly from the network without using the boot diskette. The system must support PXE. Enable the system to use PXE by using the system's BIOS setup tool or the network adapter's configuration setup Tool.

Follow the directions on screen until the Current Boot Parameters menu is displayed.

2. At the OK prompt, boot the machine to single-user state.

```
OK b -s
```

3. If necessary, check the integrity of the file system for the fallback boot environment `root (/)`.

```
# fsck mount_point
```

```
mount_point
```

A `root` file system that is known and reliable

4. Mount the active boot environment root slice to some directory (such as `/mnt/sbin`):

```
# mount logical_device_name /mnt/sbin
```

logical_device_name Any device, including a Veritas VxVM volume or Solstice DiskSuite metadvice

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 logical_device_name /mnt/sbin
```

logical_device_name Any device, including a Veritas VxVM volume or Solstice DiskSuite metadvice

7. Reboot.

```
# init 6
```

The previous working boot environment becomes the active boot environment.

Maintaining Solaris Live Upgrade Boot Environments

This chapter explains various maintenance tasks such as keeping a boot environment file system up to date or deleting a boot environment. This chapter contains the following sections:

- “Overview of Solaris Live Upgrade Maintenance ” on page 57
- “Displaying the Status of All Boot Environments” on page 58
- “Updating a Previously Configured Boot Environment” on page 60
- “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 62
- “Comparing Boot Environments” on page 62
- “Deleting an Inactive Boot Environment” on page 64
- “Displaying the Name of the Active Boot Environment” on page 65
- “Changing the Name of a Boot Environment” on page 66
- “Viewing the Configuration of a Boot Environment” on page 67

Overview of Solaris Live Upgrade Maintenance

TABLE 5-1 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 58
	Compare the active and inactive boot environments	“Comparing Boot Environments” on page 62

TABLE 5-1 Overview of Solaris Live Upgrade Maintenance *(Continued)*

Task	Description	For Instructions
	Display the name of the active boot environment	"Displaying the Name of the Active Boot Environment" on page 65
	View the configurations of a boot environment	"Viewing the Configuration of a Boot Environment" on page 67
(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 60
(Optional) Other tasks	Delete a boot environment	"Deleting an Inactive Boot Environment" on page 64
	Change the name of a boot environment	"Changing the Name of a Boot Environment" on page 66
	Cancel scheduled jobs	"Canceling a Scheduled Create, Upgrade, or Copy Job" on page 62

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 if all scheduled copy or creation operations are finished and 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 there is a copy operation in process or scheduled for a boot environment, that boot environment is considered incomplete.
- Active – Indicates if this is the active boot environment.
- ActiveOnReboot – Indicates if the boot environment becomes active on next reboot of the system.

- CopyStatus – Indicates if the creation or copy of the boot environment is scheduled, active, or in the process of being upgraded. A status of SCHEDULED prevents you performing live upgrade copy, rename, or upgrade operations.

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

- From the main menu, select Status.

A table similar to the following is displayed:

BE_name	Complete	Active	ActiveOnReboot	CopyStatus
disk_a_S7	yes	yes	yes	-
disk_b_S7database	yes	no	no	SCHEDULED
disk_b_S8	no	no	no	-

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 pending.

▼ To Display the Status of All Boot Environments (Command-Line Interface)

1. Log in as superuser.
2. Type:

```
# lustatus BE_name
```

<i>BE_name</i>	Specifies the name of the inactive boot environment to view status. If <i>BE_name</i> is omitted, <code>lustatus</code> displays status for all boot environments in the system.
----------------	--

In this example, the status for all boot environments is displayed.

```
# lustatus
BE_name          Complete  Active  ActiveOnReboot  CopyStatus
-----
disk_a_S7        yes      yes     yes             -
disk_b_S7database yes      no      no              SCHEDULED
disk_b_S8        no       no      no              -
```

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 pending.

Updating a Previously Configured Boot Environment

You can update the contents of a previously configured boot environment with the Copy menu or `lumake (1M)`. File systems from the active (source) boot environment are copied to the target boot environment and the data on the target is 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 58 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 62.

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

1. From the main menu, select Copy.
2. Type the name of the inactive boot environment to update:

```
Name of Target Boot Environment: solaris8
```

3. Continue or schedule the copy to occur later:

- To continue with the copy, press Return.
The inactive boot environment is updated.
- To schedule the copy for later, type `y`, a time (by using the `at` command format), and the email address to which to send the results:

```
Do you want to schedule the copy? y
Enter the time in 'at' format to schedule copy: 8:15 PM
Enter the address to which the copy log should be mailed:
someone@anywhere.com
```

For information on time formats, see the `at(1)` man page.

The inactive boot environment is updated.

To cancel a scheduled copy, see “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 62.

▼ To Update a Previously Configured Boot Environment (Command-Line Interface)

This procedure copies source files over outdated files on a boot environment that was created previously.

1. **Log in as superuser.**

2. **Type:**

```
# lumake -t time -s source_BE -n BE_name -m email_address
```

-t <i>time</i>	(Optional) Set up a batch job to copy over the 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, <code>at(1)</code> .
-n <i>BE_name</i>	Specifies the name of the boot environment whose file systems are to be replaced.
-s <i>source_BE</i>	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, <code>lumake</code> uses the current boot environment as the source.
-m <i>email_address</i>	Enables you to email <code>lumake</code> output to a specified address on command completion. <i>email_address</i> is not checked. You can use this option only in conjunction with <code>-t</code> .

EXAMPLE 5-1 Updating a Previously Configured Boot Environment (Command-Line Interface)

In this example, file systems from `first_disk` are copied to `second_disk` at 8:15 p.m. When the job is completed, an email is sent to Joe at `anywhere.com`.

```
# lumake -t 8:15 PM -s first_disk -n second_disk -m joe@anywhere.com
```

The copy occurs at 8:15 PM and email is sent for notification. To cancel a scheduled copy, see “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 62.

Canceling a Scheduled Create, Upgrade, or Copy Job

A boot environment's scheduled creation, upgrade, or copy job can be canceled up to the time the job starts. A job can be scheduled for a specific time 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 `lucreate (1M)` command. There can be only one job scheduled at a time on a system.

▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Character Interface)

1. From the main menu, select **Cancel**.
2. To view a list of boot environments available for canceling, press **F2**.
3. Select the boot environment to cancel.
The job no longer executes at the time specified.

▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Command-Line Interface)

1. Log in as superuser.
2. Type:

```
# lucancel
```

The job no longer executes at the time that is specified.

Comparing Boot Environments

Use the Compare menu or `lucompare` to check for differences between the active boot environment and other boot environments. To make a comparison, the inactive boot environment must be in a complete state and cannot have a copy job that is pending. See "Displaying the Status of All Boot Environments" on page 58.

The specified boot environment cannot have any partitions that are mounted with `lumount(1M)` or `mount(1M)`.

▼ To Compare Boot Environments (Character Interface)

1. From the main menu, select Compare.
2. Select either Compare to Original or Compare to an Active Boot Environment.
3. Press F3.
4. Type the names of the original (active) boot environment, the inactive boot environment, and the path to a file:

```
Name of Parent: solaris8
Name of Child: solaris8-1
Full Pathname of the file to Store Output: /tmp/compare
```

5. To save to the file, press F3.

The Compare menu displays the following file attributes:

- Mode
 - Number of links
 - Owner
 - Group
 - Checksum - Computes checksums only if the file in the specified boot environment matches its counterpart on the active boot environment in all of the fields that are described previously. If everything matches but the checksums differ, the differing checksums are appended to the entries for the compared files.
 - Size
 - Existence of files in only one boot environment
6. To return to the Compare menu, press F3.

▼ To Compare Boot Environments (Command-Line Interface)

1. Log in as superuser.
2. Type:

```
# /usr/sbin/lucompare -i infile (or) -t -o outfile BE_name
```

<code>-i infile</code>	Compare files that are listed in <i>infile</i> . The files to be compared should be an absolute file name. If the entry in the file is a directory, then comparison is recursive to the directory. Use either this option or <code>-t</code> , not both.
<code>-t</code>	Compare only nonbinary files. This comparison uses the <code>file(1)</code> command on each text file in the file system. Use either this option or <code>-i</code> , not both options.
<code>-o outfile</code>	Redirect the output of differences to <i>outfile</i> .
<i>BE_name</i>	Specifies the name of the boot environment that is compared to the active boot environment.

EXAMPLE 5-2 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 one that is not participating in an operation that will change its status. See “Displaying the Status of All Boot Environments” on page 58 for procedures to determine a boot environment’s status. Also, you cannot delete a boot environment that has file systems mounted with `lumount(1M)`.

▼ To Delete an Inactive Boot Environment (Character Interface)

1. From the main menu, select Delete.
2. Type the name of the inactive boot environment you want to delete:

```
Name of boot environment: solaris8
```

```
The inactive boot environment is deleted.
```

▼ To Delete an Inactive Boot Environment (Command-line Interface)

1. Log in as superuser.
2. Type:

```
# ludelete BE_name
```

<i>BE_name</i>	Specifies the name of the inactive boot environment that is to be deleted.
----------------	--

EXAMPLE 5-3 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 58 to determine a boot environment's status.

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

- From the main menu, select Current.

The active boot environment's name or the message "No Boot Environments are defined" is displayed.

▼ To Display the Name of the Active Boot Environment (Command-Line Interface)

- Type:

```
# /usr/sbin/lucurr -m mount_point
```

-m *mount_point*

Returns the name of the boot environment that owns *mount_point*. The name can be a mount point of the current boot environment or another boot environment. If the latter is true, the file system of the boot environment must have been mounted with `lumount(1M)` or `mount(1M)` before you entered this option.

EXAMPLE 5-4 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 -m /
```

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. 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 can contain only single-byte, 8-bit characters. Also, the new name must adhere to the following:

- 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(1M)`.
- 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 58 to determine a boot environment's status. You cannot rename a boot environment that has file systems mounted with `lumount(1M)` or `mount(1M)`.

▼ To Change the Name of an Inactive Boot Environment (Character Interface)

1. From the main menu, select **Rename**.
2. Type the boot environment to rename and then the new name.
3. To save your changes, press **F3**.

▼ To Change the Name of an Inactive Boot Environment (Command-Line Interface)

1. Log in as superuser.
2. Type:

```
# lurename -e BE_name -n new_name
```

<code>-e <i>BE_name</i></code>	Specifies the inactive boot environment name to be changed.
<code>-n <i>new_name</i></code>	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
```

Viewing the Configuration of a Boot Environment

Use the **List** menu or the `lufslist` command to list the configuration of a boot environment. The output contains the disk slice (file system), file system type, and file system size for each boot environment mount point.

▼ To View the Configuration of Each Inactive Boot Environment (Character Interface)

1. From the main menu, select **List**.
2. To view the status of a boot environment, type the name.

Name of Boot Environment: **solaris8**

3. Press F3.

The following is an example of 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 a Boot Environment (Command-Line Interface)

1. Log in as superuser.

2. Type:

```
# lufslist
```

<i>BE_name</i>	Specifies the name of the boot environment to view file system specifics.
----------------	---

The following is an example of 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

Note – In this example, you could not perform copy, rename, or upgrade operations on `disk_b_S8`, because the operation is not complete, nor on `disk_b_S7database`, because a live upgrade operation is pending.

Solaris Live Upgrade Reference

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

TABLE 6-1 Solaris Live Upgrade Command-Line Options

Task	Command
Activate an inactive boot environment.	luactivate
Cancel a scheduled copy or create job.	lucancel
Compare an active boot environment with an inactive boot environment.	lucompare
Recopy file systems to update an inactive boot environment.	lucopy
Create a boot environment.	lucreate
Name the active boot environment.	lucurr
Delete a boot environment.	ludelete
List system file systems for each boot environment.	lufslist
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 not active.	lumount
Rename a boot environment.	lurename
List status of all boot environments.	lustatus
Upgrade an operating environment or install a flash archive an inactive boot environment.	luupgrade

TABLE 6-1 Solaris Live Upgrade Command-Line Options *(Continued)*

Task	Command
Enable an unmount 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 not active.	luumount

Troubleshooting

This chapter contains solutions for problems you might encounter when upgrading using Solaris Live Upgrade.

Upgrade Problems

Cannot Use the Solaris 8 Device Configuration Assistant *Intel Platform Edition* diskette to Boot the New Boot Environment

Mount Failed. Unable to mount a Solaris root file system from the device: DISK:

Description: On an Intel system, when using the Device Configuration Assistant diskette and after upgrading and activating the new boot environment, the activation fails.

When booting from the Solaris 8 Device Configuration Assistant *Intel Platform Edition* diskette, the diskette does not find the correct path to partition with the root (/) file system. You must manually update the boot diskette before activating the new boot environment. For the procedure on updating the boot diskette, see “x86: (Optional) Update Boot Diskette Before Activating” on page 49.

Creation of New Boot Environment Fails When Reconfiguring Swap

ERROR: Template filesystem definition failed for -..

ERROR: Configuration of BE failed.

Cause: The swap slice is being used by a boot environment other than the current boot environment or the source boot environment specified by the `-s` option and the creation of a new boot environment fails.

Description: 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 slice contains a swap, ufs, or any other file system.

Solution: Choose a slice for swap following the above guidelines.

System Panics When Upgrading On Veritas VxVm

When using Solaris Live Upgrade while upgrading and running Veritas VxVM, the system panics on reboot unless you upgrade by using the following procedure. The problem occurs if packages do not conform to Solaris advanced packaging guidelines.

1. Create an inactive boot environment. See “Creating a New Boot Environment” on page 27.
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 vfstab.novxfs > vfstab < '/vx\/dsk\/s\/^\/#\/g'
```

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: system.novxfs > system < drv\/vx\/s\/^\/*/'
```

The first character of each line is changed to `*`, which makes the line a command line. Note this 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 4.

- 4. Activate the inactive boot environment. See "Activating a Boot Environment" on page 49.

- 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 containing "vxvm" or "VXVM" are displayed that can be ignored. The inactive boot environment becomes active.

- 7. Upgrade Veritas.

- a. Remove the Veritas `VRTSvmsa` package from system, for example:

```
# pkgrm VRTSvmsa
```

- b. Change directories to the Veritas packages.

```
# cd /location_of_Veritas_software
```

c. Add the latest Veritas packages to the system:

```
# pkgadd -d `pwd` VRTSvxvm VRTSvmsa VRTSvmdoc VRTSvmman VRTSvmdev
```

8. Restore the original `vfstab` and system files:

```
# cp /etc/vfstab.original /etc/vfstab  
# cp /etc/system.original /etc/system
```

9. Reboot the system.

```
# init 6
```

Glossary

active boot environment	The boot environment that is currently booted.
archive	A file that contains all of the files that were copied from a master system. The file also contains identification information about the archive, such as name and the date you created the archive. When you select to install an archive on a system, the system then contains the exact configuration of the master system you used to create the archive.
arrow keys	One of the four directional keys on the numeric keypad.
boot	To load the system software into memory and start it.
boot environment	A bootable Solaris environment that consists of a set of disk slices and associated mount points and file systems. These disk slices might be on the same disk or distributed across multiple disks.
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.
clone system	A system that you installed by using a Web Start Flash archive. The clone system has the exact same installation configuration as the master system.
command line	A string of characters that begin with a command, often followed by arguments, including options, file names, and other expressions, and terminated by the end-of-line character.
critical file systems	File systems that are required by the Solaris operating environment and are separate mount points in the <code>vfstab</code> of the active and inactive boot environments. Examples are <code>root (/) /usr, /var</code> or <code>/opt</code> . These file systems are always copied from the source to the inactive boot environment.

disk	A round platter, or set of platters, of a magnetized medium organized into concentric tracks and sectors for storing data such as files.
/export	A file system on an OS server that is shared with other systems on a network. For example, the <code>/export</code> 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 <code>/export</code> 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 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 IA based systems. To install the Solaris software, you must set up at least one Solaris 8 <code>fdisk</code> partition on an IA based system. IA based systems allow up to four different <code>fdisk</code> partitions on a disk. These partitions can be used to hold individual operating systems. Each operating system must be located on a unique <code>fdisk</code> partition. A system can only have one Solaris <code>fdisk</code> partition per disk.
file system	In the SunOS™ operating system, a tree-structured network of files and directories that you can access.
format	To put data into a structure or divide a disk into sectors for receiving data.
function key	One of the 10 or more keyboard keys that are labeled F1, F2, F3, and so on that are mapped to particular tasks.
hard link	A directory entry that references a file on disk. More than one such directory entry can reference the same physical file.
HTTP	(Hypertext Transfer Protocol) (n.) The Internet protocol that fetches hypertext objects from remote hosts. This protocol is based on TCP/IP.
inactive boot environment	Any boot environment that is not currently booted and not designated for activation on next reboot.
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 using a profile server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files.
JumpStart installation	A type of installation in which the Solaris 8 software is automatically installed on a system by using the factory-installed JumpStart software.
master system	A system that you use to create a Web Start Flash archive. The system configuration is saved in the archive.

metadevice	See volume.
mount	The process of accessing a directory from a disk that is attached to a machine making the mount request or 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, <code>/usr</code>).
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.
network installation	A way to install software over the network—from a system with a CD-ROM drive to a system without a CD-ROM drive. Network installations require a <i>name server</i> and an <i>install server</i> .
networked systems	A group of systems (called hosts) that are connected through hardware and software, so they can communicate and share information; referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.
non-networked systems	Systems that are not connected to a network or do not rely on other systems.
/opt	A file system that contains the mount points for third-party and other software.
package	A collection of software that is grouped into a single entity for modular installation. The Solaris 8 software is divided into <i>software groups</i> , which are each composed of <i>clusters</i> and packages.
profile	A text file that defines how to install the Solaris software (for example, 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 <i>rules file</i> .
profile diskette	A diskette that contains all the essential custom JumpStart files in its root directory (JumpStart directory).
/ (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. 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.
shareable file systems	File systems that are user-defined files such as <code>/export/home</code> , and <code>/swap</code> that are shared between the active and inactive boot environment. Shareable file systems contain the same mount point in the <code>vfstab</code> 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.
Solaris 8 CD images	The Solaris 8 software that is installed on a system, which you can access on the Solaris 8 CDs or an install server's hard disk to which you have copied the Solaris 8 CD images.
swap space	A slice or file that temporarily holds the contents of a memory area till it can be loaded back into memory. Also known as the <code>/swap</code> or <code>swap</code> file system.
unmount	The process of removing access to a directory on a disk that is attached to a machine or a remote disk on a network.
/usr	A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large <code>/usr</code> file system with a server rather than maintaining a local copy minimizes the overall disk space required to install and run the Solaris 8 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 contain system files that are likely to change or grow over the life of the system. These include system logs, <code>vi</code> files, mail files, and <code>uucp</code> files.
volume	<p>A group of physical slices accessed as a single logical device by concatenation, striping, mirroring, setup RAID5 volumes, or logging physical devices. After they are created, volumes are used like slices. The volume maps logical block address to the correct location on one of the physical devices. The type of mapping depends on the configuration of the particular volume.</p> <p>Also known as pseudo, or virtual device in standard UNIX terms.</p>
Volume Manager	A program that provides a mechanism to administer and obtain access to the data on CD-ROMs and diskettes.

Index

B

boot environment, view status, 67

C

cancel a Live Upgrade job, 62
change name of Live Upgrade boot environment, 66
commands for Live Upgrade, 69
compare Live Upgrade boot environments, 62
configuring Live Upgrade file systems, 29
copy, Live Upgrade file systems, 60
creating, a boot environment for Live Upgrade, 27

D

delete a Live Upgrade boot environment, 64
disk space, for Solaris Live Upgrade, 20
display name of Live Upgrade boot environment, 65

F

failed upgrade, recovery with Solaris Live Upgrade, 52
files and file systems
 copying a shareable file system for Live Upgrade, 38
 customizing for Live Upgrade, 29

L

Live Upgrade, *See* Solaris Live Upgrade

P

planning
 disk space for Live Upgrade, 20
 for Solaris Live Upgrade, 19
 system requirements for Live Upgrade, 19

R

requirements, to use Live Upgrade, 19

S

slices
 customizing file systems for Live Upgrade, 29
 guidelines for Live Upgrade, 21
Solaris Live Upgrade
 Activating a boot environment, 49
 cancel a job, 62
 change name of a boot environment, 66
 commands, 69
 compare boot environments, 62
 configuring file systems, 29
 copying a shareable file system, 38
 creating a boot environment, 27
 delete a boot environment, 64

Solaris Live Upgrade (Continued)

- display name of boot environment, 65
- failed upgrade recovery, 52
- file system slices, 29
- installing, 25
- installing a Web Start Flash archive, 46
- overview, 12
- printing to a file, 29
- starting, 26
- stopping, 26
- upgrade task map, 42
- upgrading a boot environment, 42
- view configuration of boot environments, 67
- status, display boot environment, 58
- swap file systems
 - customizing for Live Upgrade (character interface), 31
 - customizing for Live Upgrade (command line), 36

U

- upgrade, with Solaris Live Upgrade, 42
- upgrade installation, failed upgrade recovery for Live Upgrade, 52

W

- Web Start Flash, installing on Live Upgrade boot environment, 46