

Oracle® Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Preface	11
Part I Upgrading With Live Upgrade	15
1 Where to Find Oracle Solaris Installation Planning Information	17
Where to Find Planning and System Requirement Information	17
2 Live Upgrade (Overview)	19
Live Upgrade Introduction	19
Live Upgrade Process	20
Creating a Boot Environment	22
Creating a Boot Environment With RAID-1 Volume File Systems	27
Upgrading a Boot Environment	33
Activating a Boot Environment	38
Falling Back to the Original Boot Environment	39
Maintaining a Boot Environment	41
3 Live Upgrade (Planning)	43
Live Upgrade Requirements	43
Live Upgrade System Requirements	43
Installing Live Upgrade	44
Live Upgrade Disk Space Requirements	46
Live Upgrade Requirements If Creating RAID-1 Volumes (Mirrors)	47
Upgrading a System With Packages or Patches	47
Guidelines for Creating File Systems With the <code>lucreate</code> Command	48
Guidelines for Selecting Slices for File Systems	49
Guidelines for Selecting a Slice for the root (<code>/</code>) File System	49

Guidelines for Selecting Slices for Mirrored File Systems	50
Guidelines for Selecting a Slice for a Swap Volume	51
Guidelines for Selecting Slices for Shareable File Systems	52
Customizing a New Boot Environment's Content	53
Synchronizing Files Between Boot Environments	54
Adding Files to the /etc/lu/syncclist File	54
Forcing a Synchronization Between Boot Environments	55
Booting Multiple Boot Environments	56
Live Upgrade Character User Interface	57
4 Using Live Upgrade to Create a Boot Environment (Tasks)	59
Installing Live Upgrade and Creating Boot Environments (Task Map)	59
Installing Live Upgrade	60
▼ How to Install Live Upgrade With the pkgadd Command	60
▼ How to Install Live Upgrade With the Oracle Solaris Installation Program	61
Installing Patches Needed by Live Upgrade	62
Creating a New Boot Environment	64
▼ How to Create a Boot Environment for the First Time	65
▼ How to Create a Boot Environment and Merge File Systems	67
▼ How to Create a Boot Environment and Split File Systems	68
▼ How to Create a Boot Environment and Reconfiguring Swap	70
▼ How to Create a Boot Environment and Reconfigure Swap by Using a List	72
▼ How to Create a Boot Environment and Copy a Shareable File System	74
▼ How to Create a Boot Environment From a Different Source	75
▼ How to Create an Empty Boot Environment for a Flash Archive	76
▼ How to Create a Boot Environment With RAID-1 Volumes (Mirrors)	78
Creating a Boot Environment and Customizing the Content	83
5 Upgrading With Live Upgrade (Tasks)	85
Upgrading a Boot Environment (Task Map)	85
Upgrading a Boot Environment	86
Guidelines for Upgrading	86
▼ How to Upgrade a Network Installation Image on a Boot Environment	87
▼ How to Upgrade a Network Installation Image From Multiple CDs	88
Adding Packages to or Removing Packages From a Network Installation Image on a Boot	

Environment	90
Adding Patches to or Removing Patches From a Network Installation Image on a Boot Environment	91
Checking Packages Installed on a Boot Environment	93
Upgrading by Using a JumpStart Profile	93
JumpStart Keyword Values	98
Installing Flash Archives on a Boot Environment	100
▼ How to Install a Flash Archive on a Boot Environment	101
▼ How to Install a Flash Archive With a Profile	102
▼ How to Install a Flash Archive With a Profile Keyword	103
Activating a Boot Environment	104
Requirements and Limitations for Activating a Boot Environment	105
▼ How to Activate a Boot Environment	106
▼ How to Activate a Boot Environment and Synchronize Files	107
x86: Activating a Boot Environment With the GRUB Menu	108
▼ x86: How to Activate a Boot Environment With the GRUB Menu	109
6 Failure Recovery: Falling Back to the Original Boot Environment (Tasks)	111
SPARC: Falling Back to the Original Boot Environment	112
▼ SPARC: To Fall Back Despite Successful New Boot Environment Activation	112
▼ SPARC: To Fall Back From a Failed Boot Environment Activation	112
▼ SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image	113
x86: Falling Back to the Original Boot Environment	114
▼ x86: To Fall Back Despite Successful New Boot Environment Activation With the GRUB Menu	114
▼ x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu ...	115
▼ x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu and the DVD or CD	117
7 Maintaining Live Upgrade Boot Environments (Tasks)	121
Overview of Live Upgrade Maintenance	122
Maintenance Activities for Boot Environments	122
Displaying the Status of All Boot Environments	122
Updating a Previously Configured Boot Environment	123
Canceling a Scheduled Create, Upgrade, or Copy Job	124

Comparing Boot Environments	124
Deleting an Inactive Boot Environment	125
Displaying the Name of the Active Boot Environment	126
Changing the Name of a Boot Environment	126
Adding or Changing a Description Associated With a Boot Environment Name	127
Viewing the Configuration of a Boot Environment	128
8 Upgrading the Oracle Solaris OS on a System With Non-Global Zones Installed	131
Upgrading With Live Upgrade and Installed Non-Global Zones (Overview)	131
Understanding Oracle Solaris Zones and Live Upgrade	132
Guidelines for Using Live Upgrade With Non-Global Zones (Planning)	136
Creating a Boot Environment When a Non-Global Zone Is on a Separate File System	137
Creating and Upgrading a Boot Environment When Non-Global Zones Are Installed (Tasks)	137
▼ Upgrading With Live Upgrade When Non-Global Zones Are Installed on a System (Tasks)	138
Upgrading a System With Non-Global Zones Installed (Example)	142
Upgrading With Live Upgrade When Non-Global Zones Are Installed on a System	143
Administering Boot Environments That Contain Non-Global Zones	144
▼ To View the Configuration of a Boot Environment's Non-Global Zone File Systems	144
▼ To Compare Boot Environments for a System With Non-Global Zones Installed	145
Using the lumount Command on a System That Contains Non-Global Zones	146
9 Live Upgrade Examples	147
Example of Upgrading With Live Upgrade	147
Preparing to Use Live Upgrade	147
Using Pre Flight Checker Tool	150
To Create a Boot Environment	151
To Upgrade the Inactive Boot Environment	151
To Check if Boot Environment Is Bootable	151
To Activate the Inactive Boot Environment	151
(Optional) To Fall Back to the Source Boot Environment	152
Example of Detaching and Upgrading One Side of a RAID-1 Volume (Mirror)	156
Example of Migrating From an Existing Volume to a Solaris Volume Manager RAID-1 Volume	159

Example of Creating an Empty Boot Environment and Installing a Flash Archive	159
To Create an Empty Boot Environment	160
To Install a Flash Archive on the New Boot Environment	161
To Activate the New Boot Environment	161
Part II Upgrading and Migrating With Live Upgrade to a ZFS Root Pool	163
10 Live Upgrade and ZFS (Overview)	165
What's New in Oracle Solaris 10 8/11 Release	166
What's New in the Solaris 10 10/09 Release	166
Introduction to Using Live Upgrade With ZFS	167
Migrating From a UFS File System to a ZFS Root Pool	167
Migrating From a UFS Root (/) File System to ZFS Root Pool	168
Migrating a UFS File System With Solaris Volume Manager Volumes Configured to a ZFS Root File System	171
Creating a New Boot Environment From a ZFS Root Pool	172
Creating a New Boot Environment Within the Same Root Pool	172
Creating a New Boot Environment on Another Root Pool	174
Creating a New Boot Environment From a Source Other Than the Currently Running System	175
Creating a ZFS Boot Environment on a System With Non-Global Zones Installed	176
11 Live Upgrade for ZFS (Planning)	177
System Requirements and Limitations When Using Live Upgrade	177
12 Creating a Boot Environment for ZFS Root Pools	181
Migrating a UFS File System to a ZFS File System	181
▼ How to Migrate a UFS File System to a ZFS File System	182
Creating a Boot Environment Within the Same ZFS Root Pool	188
▼ How to Create a ZFS Boot Environment Within the Same ZFS Root Pool	188
Creating a Boot Environment In a New Root Pool	192
▼ How to Create a Boot Environment on a New ZFS Root Pool	193
Creating a Boot Environment From a Source Other Than the Currently Running System	197
Falling Back to a ZFS Boot Environment	198

13	Live Upgrade for ZFS With Non-Global Zones Installed	199
	Creating a ZFS Boot Environment on a System With Non-Global Zones Installed (Overview and Planning)	199
	Migrating From a UFS Root (/) File System With Non-Global Zones Installed to ZFS Root Pool (Tasks)	200
	▼ How to Migrate a UFS File System to a ZFS Root Pool on a System With Non-Global Zones	200
Part III	Appendices	207
A	Live Upgrade Command Reference	209
B	Troubleshooting (Tasks)	211
	Problems With Setting Up Network Installations	211
	Problems With Booting a System	212
	Error Messages When Booting From Media	212
	General Problems When Booting From Media	213
	Booting From the Network, Error Messages	214
	General Problems When Booting From the Network	217
	Initial Installation of the Oracle Solaris OS	217
	▼ x86: How to Check an IDE Disk for Bad Blocks	218
	Upgrading the Oracle Solaris OS	220
	Upgrading Error Messages	220
	General Problems When Upgrading	221
	▼ How to Continue Upgrading After a Failed Upgrade	223
	x86: Problems With Live Upgrade When You Use GRUB	223
	System Panics When Upgrading With Live Upgrade Running Veritas VxVm	225
	x86: Service Partition Not Created by Default on Systems With No Existing Service Partition	227
	▼ How to Include a Service Partition When Installing Software From a Network Installation Image or From the Oracle Solaris Operating System DVD	228
	▼ How to Include a Service Partition When Installing From the Oracle Solaris Software - 1 CD or From a Network Installation Image	228

C	Additional SVR4 Packaging Requirements (Reference)	229
	Preventing Modification of the Current OS	229
	Using Absolute Paths	229
	Using the pkgadd -R Command	230
	Differences Between \$PKG_INSTALL_ROOT and \$BASEDIR	230
	Guidelines for Writing Scripts	231
	Maintaining Diskless Client Compatibility	231
	Verifying Packages	232
	Preventing User Interaction When Installing or Upgrading	233
	Setting Package Parameters For Zones	234
	For Further Information	237
D	Using the Patch Analyzer When Upgrading (Tasks)	239
	Upgrading to an Oracle Solaris Update Release	239
	▼ How to Use the analyze_patches Script	240
	Glossary	241
	Index	249

Preface

This book describes how to install and upgrade the Oracle Solaris operating system (OS) on both networked and nonnetworked SPARC and x86 architecture based systems by using Live Upgrade, a feature of Oracle Solaris.

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

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

In this document, these x86 related terms mean the following:

- x86 refers to the larger family of 64-bit and 32-bit x86 compatible products.
- x64 relates specifically to 64-bit x86 compatible CPUs.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Oracle Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Book

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

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

Related Books

The following table lists documentation for system administrators.

Header	Header
Do you want to review information such as system requirements or high-level planning information, a high-level overview of Oracle Solaris ZFS installations, booting, Oracle Solaris Zones partitioning technology, or creating RAID-1 volumes?	Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade
Do you need to install a single system from DVD or CD media? The Oracle Solaris installation program steps you through an installation.	Oracle Solaris 10 1/13 Installation Guide: Basic Installations
Do you need to upgrade or patch your system with almost no downtime? Save system downtime when upgrading by using Live Upgrade.	Oracle Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning
Do you need to install a secure installation over the network or Internet? Use WAN boot to install a remote client. Or, do you need to install over the network from a network installation image? The Oracle Solaris installation program steps you through an installation.	Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations
Do you need to install Oracle Solaris on multiple machines? Use JumpStart, a feature of Oracle Solaris, to automate your installation.	Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations
Do you need to install or patch multiple systems quickly? Use Flash Archive software to create an archive and install a copy of the OS on clone systems.	Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation)
Do you need to back up your system?	Chapter 19, “Backing Up and Restoring UFS File Systems (Overview/Tasks),” in <i>System Administration Guide: Devices and File Systems</i>
Do you need troubleshooting information, a list of known problems, or a list of patches for this release?	Oracle Solaris Release Notes
Do you need to verify that your system works on Oracle Solaris?	SPARC: Oracle Solaris Sun Hardware Platform Guide
Do you need to check on which packages have been added, removed, or changed in this release?	Oracle Solaris Package List
Do you need to verify that your system and devices work with Solaris SPARC and x86 based systems and other third-party vendors?	Solaris Hardware Compatibility List for x86 Platforms

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Typographic Conventions

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

TABLE P-1 Typographic Conventions

Typeface	Description	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name%</code> su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows UNIX system prompts and superuser prompts for shells that are included in the Oracle Solaris OS. In command examples, the shell prompt indicates whether the command should be executed by a regular user or a user with privileges.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$

TABLE P-2 Shell Prompts *(Continued)*

Shell	Prompt
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#

PART I

Upgrading With Live Upgrade

This part provides an overview and instructions for using Live Upgrade to create and upgrade an inactive boot environment. The boot environment can then be switched to become the current boot environment. This part is written for a system with a UFS root (/) file system. However, many commands can be used for the ZFS file system.

Where to Find Oracle Solaris Installation Planning Information

This book provides information on how to use Live Upgrade, a feature of Oracle Solaris, to upgrade the Oracle Solaris operating system. This book provides all you need to know about using Live Upgrade, but a planning book in our collection of installation documentation might be useful to read before you begin. The following references provide useful information before you upgrade your system.

Where to Find Planning and System Requirement Information

The *Oracle Solaris 10 8/11 Installation Guide: Planning For Installation and Upgrade* provides system requirements and high-level planning information, such as planning guidelines for file systems, and upgrade planning and much more. The following list describes the chapters in the planning book and provides links to those chapters.

Chapter Descriptions From the Planning Guide	Reference
This chapter provides you with information about decisions you need to make before you install or upgrade the Oracle Solaris OS. Examples are deciding when to use a network installation image or DVD media and descriptions of all the Oracle Solaris installation programs.	Chapter 2, “Oracle Solaris Installation and Upgrade Roadmap,” in <i>Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</i>
This chapter describes system requirements to install or upgrade to the Oracle Solaris OS. General guidelines for planning the disk space and default swap space allocation are also provided. Upgrade limitations are also described.	Chapter 3, “System Requirements, Guidelines, and Upgrade Information,” in <i>Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</i>
This chapter contains checklists to help you gather all of the information that you need to install or upgrade your system. This information is useful, for example, if you are performing an interactive installation. You'll have all the information in the checklist that you'll need to do an interactive installation.	Chapter 4, “Gathering Information Before an Installation or Upgrade,” in <i>Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</i>

Chapter Descriptions From the Planning Guide	Reference
These chapters provide overviews of several technologies that relate to Oracle Solaris OS installation or upgrade. Guidelines and requirements related to these technologies are also included. These chapters include information about ZFS installations, booting, Oracle Solaris Zones partitioning technology, and RAID-1 volumes that can be created at installation.	Part II, “Understanding Installations Related to ZFS, Booting, Oracle Solaris Zones, and RAID-1 Volumes,” in <i>Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</i>

Live Upgrade (Overview)

This chapter describes the process used by Live Upgrade, a feature of Oracle Solaris.

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

Live Upgrade Introduction

Note – This chapter describes Live Upgrade for UFS file systems. For an overview of migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see [Chapter 10, “Live Upgrade and ZFS \(Overview\)”](#).

Live Upgrade provides a method of upgrading a system while the system continues to operate. While your current boot environment is running, you can duplicate the boot environment, then upgrade the duplicate. Or, rather than upgrading, you can install a flash archive on a boot environment. The original system configuration remains fully functional and unaffected by the upgrade or installation of an archive. When you are ready, you can activate the new boot environment by rebooting the system. If a failure occurs, you can quickly revert to the original boot environment with a simple reboot. This switch eliminates the normal downtime of the test and evaluation process.

Live Upgrade enables you to duplicate a boot environment without affecting the currently running system. You can then do the following:

- Upgrade a system.
- Change the current boot environment's disk configuration to different file system types, sizes, and layouts on the new boot environment.

- Maintain numerous boot environments with different images. For example, you can create one boot environment that contains current patches and create another boot environment that contains an Update release.

Tip – You can use the Live Upgrade Pre Flight Check tool to detect system configuration issues that would likely cause a Live Upgrade operation to fail. This tool needs to be run before invoking the `lucreate` command to create a boot environment. For more information about the Live Upgrade Pre Flight Check tool, see the [lupc\(1M\)](#) man page.

For more information on examples using the Pre Flight Check tool, see “[Using Pre Flight Checker Tool](#)” on page 150.

Some understanding of basic system administration is necessary before using Live Upgrade. For background information about system administration tasks such as managing file systems, mounting, booting, and managing swap, see the *[System Administration Guide: Devices and File Systems](#)*.

Live Upgrade Process

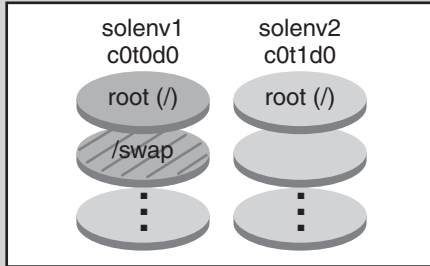
The following overview describes the tasks necessary to create a copy of the current boot environment, upgrade the copy, and switch the upgraded copy to become the active boot environment. The fallback process of switching back to the original boot environment is also described. [Figure 2–1](#) describes this complete Live Upgrade process.

FIGURE 2-1 Live Upgrade Process

Solaris Live Upgrade Process

① Create a boot environment.

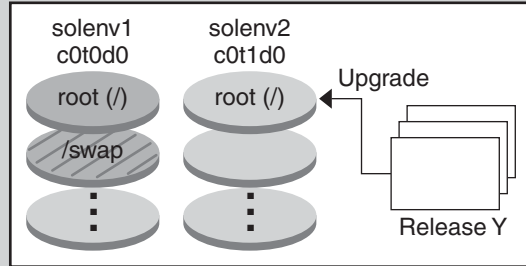
```
# lucreate -c solenv1 \
-m /dev/dsk/c0t1d0s0:ufs \
-n solenv2
```



② Upgrade an inactive boot environment.

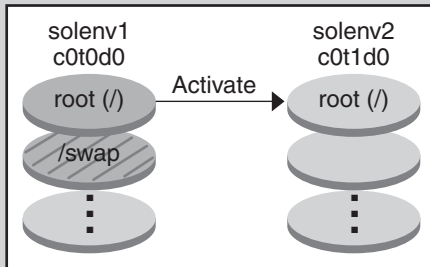
For a standard upgrade:

```
(a) # luupgrade -u -n solenv2 \
-s /net/installmachine/export/Solaris/OS_image
```



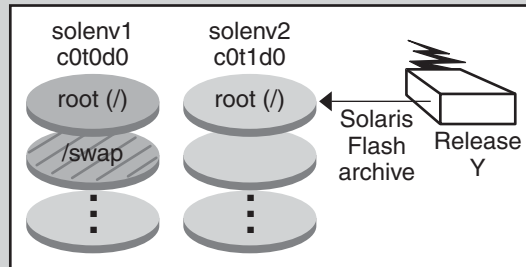
③ Activate the inactive boot environment with a reboot.

```
# luactivate solenv2
# init 6
```



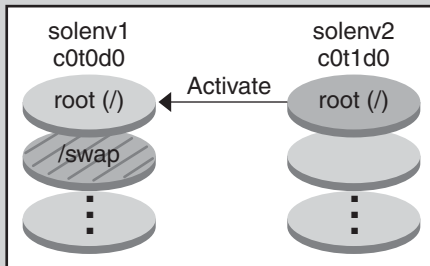
For a Solaris Flash archive:

```
(b) # luupgrade -f -n solenv2 \
-s /net/installmachine/export/Solaris/Release_Y \
-a /net/server/archive/Release_Y
```



④ (Optional) Fallback to the original boot environment.

```
# luactivate solenv1
# init 6
```



⑤ (Optional) Remove the inactive boot environment.

```
# ludelete solenv2
```

The following sections describe the Live Upgrade process.

- 1. A new boot environment can be created on a physical slice or a logical volume:
 - [“Creating a Boot Environment” on page 22](#)
 - [“Creating a Boot Environment With RAID-1 Volume File Systems” on page 27](#)
- 2. [“Upgrading a Boot Environment” on page 33](#)
- 3. [“Activating a Boot Environment” on page 38](#)
- 4. [“Falling Back to the Original Boot Environment” on page 39](#)

Creating a Boot Environment

The process of creating a boot environment provides a method of copying critical file systems from an active boot environment to a new boot environment. The disk is reorganized if necessary, file systems are customized, and the critical file systems are copied to the new boot environment.

File System Types

Live Upgrade distinguishes between two file system types: critical file systems and shareable. The following table describes these file system types.

File System Type	Description	Examples and More Information
Critical file systems	Critical file systems are required by the Oracle Solaris OS. These file systems are separate mount points in the <code>vfstab</code> of the active and inactive boot environments. These file systems are always copied from the source to the inactive boot environment. Critical file systems are sometimes referred to as <i>nonshareable</i> .	Examples are <code>root (/)</code> , <code>/usr</code> , <code>/var</code> , or <code>/opt</code> .
Shareable file systems	Shareable file systems are user-defined files such as <code>/export</code> that contain the same mount point in the <code>vfstab</code> in both the active and inactive boot environments. Therefore, updating shared files in the active boot environment also updates data in the inactive boot environment. When you create a new boot environment, shareable file systems are shared by default. But you can specify a destination slice and then the file systems are copied.	<code>/export</code> is an example of a file system that can be shared. For more detailed information about shareable file systems, see “Guidelines for Selecting Slices for Shareable File Systems” on page 52 .

File System Type	Description	Examples and More Information
Swap	<ul style="list-style-type: none">■ For UFS file systems, swap is a special shareable volume. Like a shareable file system, all swap slices are shared by default. But, if you specify a destination directory for swap, the swap slice is copied.■ For ZFS file systems, swap and dump volumes are shared within the pool.	<ul style="list-style-type: none">■ For procedures about reconfiguring swap for UFS file systems, see “How to Create a Boot Environment and Reconfiguring Swap” on page 70.■ For information about swap for ZFS root pools, see “System Requirements and Limitations When Using Live Upgrade” on page 177

Creating RAID-1 Volumes on File Systems

Live Upgrade can create a boot environment with RAID-1 volumes (mirrors) on file systems. For an overview, see [“Creating a Boot Environment With RAID-1 Volume File Systems”](#) on page 27.

Copying File Systems

The process of creating a new boot environment begins by identifying an unused slice where a critical file system can be copied. If a slice is not available or a slice does not meet the minimum requirements, you need to format a new slice.

After the slice is defined, you can reconfigure the file systems on the new boot environment before the file systems are copied into the directories. You reconfigure file systems by splitting and merging them, which provides a simple way of editing the `vfstab` to connect and disconnect file system directories. You can merge file systems into their parent directories by specifying the same mount point. You can also split file systems from their parent directories by specifying different mount points.

After file systems are configured on the inactive boot environment, you begin the automatic copy. Critical file systems are copied to the designated directories. Shareable file systems are not copied, but are shared. The exception is that you can designate some shareable file systems to be copied. When the file systems are copied from the active to the inactive boot environment, the files are directed to the new directories. The active boot environment is not changed in any way.

For procedures to split or merging file systems	<ul style="list-style-type: none">■ “How to Create a Boot Environment and Merge File Systems” on page 67■ “How to Create a Boot Environment and Split File Systems” on page 68
For an overview of creating a boot environment with RAID-1 volume file systems	“Creating a Boot Environment With RAID-1 Volume File Systems” on page 27

Examples of Creating a New Boot Environment

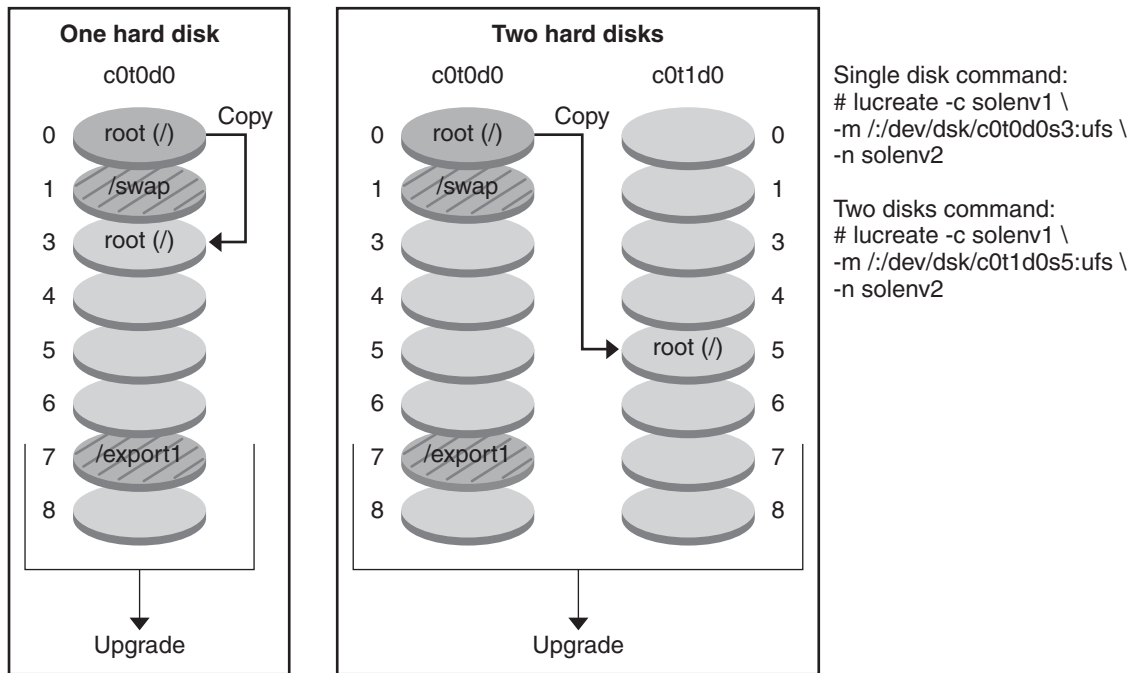
For UFS file systems, the figures in this section illustrate various ways of creating new boot environments.

For ZFS file system information, see [Chapter 10, “Live Upgrade and ZFS \(Overview\)”](#)

The following figure shows that critical file system root (/) has been copied to another slice on a disk to create a new boot environment. The active boot environment contains the root (/) file system on one slice. The new boot environment is an exact duplicate with the root (/) file system on a new slice. The /swap volume and /export/home file system are shared by the active and inactive boot environments.

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

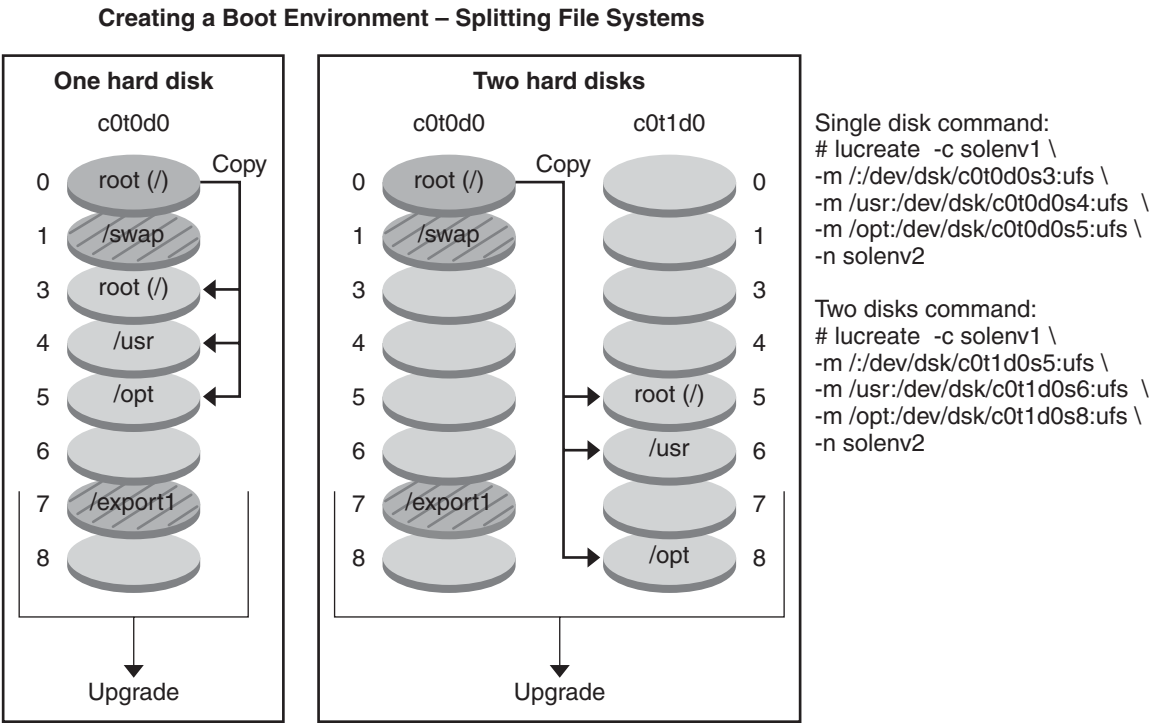
Creating a Boot Environment – Copying the root (/) File System to a single slice



- Current release X
Critical file system root (/)
- Inactive release X
Critical file systems root (/)
- Shared file systems

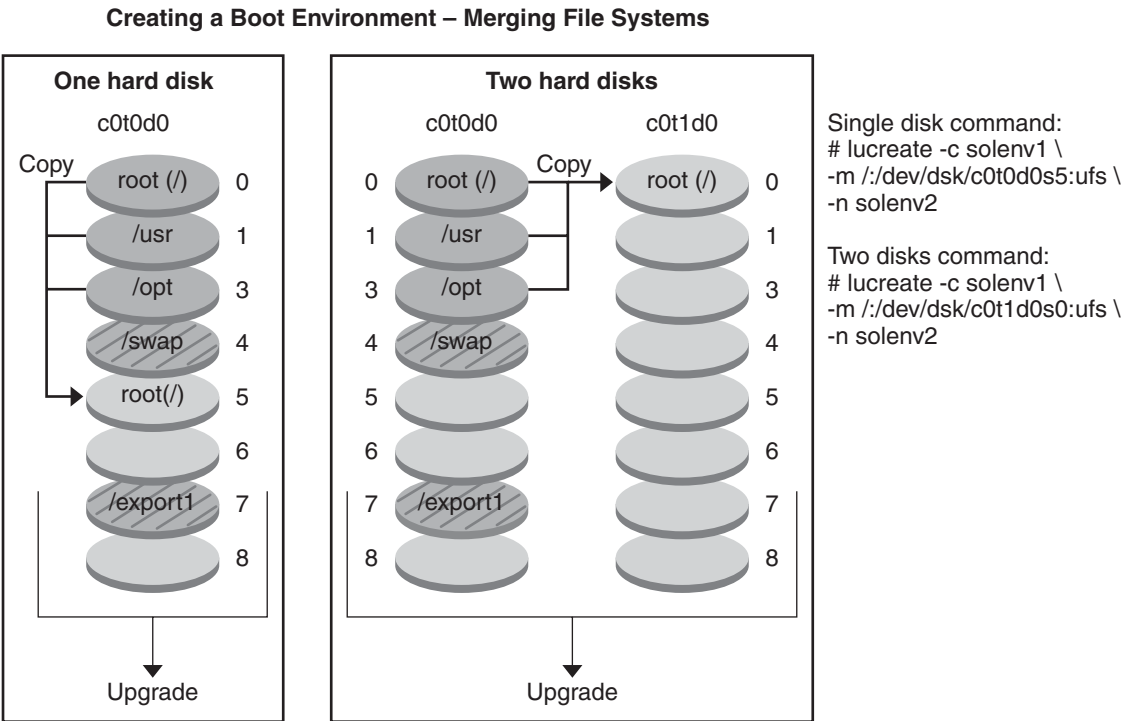
The following figure shows critical file systems that have been split and have been copied to slices on a disk to create a new boot environment. The active boot environment contains the root (/) file system on one slice. On that slice, the root (/) file system contains the /usr, /var, and /opt directories. In the new boot environment, the root (/) file system is split and /usr and /opt are put on separate slices. The /swap volume and /export/home file system are shared by both boot environments.

FIGURE 2-3 Creating an Inactive Boot Environment – Splitting File Systems



The following figure shows critical file systems that have been merged and have been copied to slices on a disk to create a new boot environment. The active boot environment contains the root (/) file system, /usr, /var, and /opt with each file system on their own slice. In the new boot environment, /usr and /opt are merged into the root (/) file system on one slice. The /swap volume and /export/home file system are shared by both boot environments.

FIGURE 2-4 Creating an Inactive Boot Environment – Merging File Systems



- Current release X
Critical file systems root (/) /usr /opt
- Inactive release Y
Critical file systems root (/)
- Shared file systems

Creating a Boot Environment With RAID-1 Volume File Systems

Live Upgrade uses Solaris Volume Manager technology to create a boot environment that can contain file systems encapsulated in RAID-1 volumes. Solaris Volume Manager provides a powerful way to reliably manage your disks and data by using volumes. Solaris Volume Manager enables concatenations, stripes, and other complex configurations. Live Upgrade enables a subset of these tasks, such as creating a RAID-1 volume for the root (/) file system.

A volume can group disk slices across several disks to transparently appear as a single disk to the OS. Live Upgrade is limited to creating a boot environment for the root (/) file system that

contains single-slice concatenations inside a RAID-1 volume (mirror). This limitation is because the boot PROM is restricted to choosing one slice from which to boot.

Managing Volumes With Live Upgrade

When creating a boot environment, you can use Live Upgrade to manage the following tasks.

- Detach a single-slice concatenation (submirror) from a RAID-1 volume (mirror). The contents can be preserved to become the content of the new boot environment if necessary. Because the contents are not copied, the new boot environment can be quickly created. After the submirror is detached from the original mirror, the submirror is no longer part of the mirror. Reads and writes on the submirror are no longer performed through the mirror.
- Create a boot environment that contains a mirror.
- Attach a maximum of three single-slice concatenations to the newly created mirror.

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

Note – If VxVM volumes are configured on your current system, the `lucreate` command can create a new boot environment. When the data is copied to the new boot environment, the Veritas file system configuration is lost and a UFS file system is created on the new boot environment.

For more information, see the following resources:

- For step-by-step procedures, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)”](#) on page 78.
- For an overview of creating RAID-1 volumes when installing, see [Chapter 8, “Creating RAID-1 Volumes \(Mirrors\) During Installation \(Overview\)”](#), in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*.
- For in-depth information about other complex Solaris Volume Manager configurations that are not supported if you are using the Live Upgrade, see [Chapter 2, “Storage Management Concepts”](#), in *Solaris Volume Manager Administration Guide*.

Mapping Solaris Volume Manager Tasks to Live Upgrade

Live Upgrade manages a subset of Solaris Volume Manager tasks. The following table shows the Solaris Volume Manager components that the Live Upgrade can manage.

TABLE 2-1 Classes of Volumes

Term	Description
Concatenation	A RAID-0 volume. If slices are concatenated, the data is written to the first available slice until that slice is full. When that slice is full, the data is written to the next slice, serially. A concatenation provides no data redundancy unless it is contained in a mirror.
Mirror	A RAID-1 volume. See RAID-1 volume.
RAID-1 volume	A class of volume that replicates data by maintaining multiple copies. A RAID-1 volume is sometimes called a <i>mirror</i> . A RAID-1 volume is composed of one or more RAID-0 volumes that are called submirrors.
RAID-0 volume	A class of volume that can be a stripe or a concatenation. These components are also called <i>submirrors</i> . A stripe or concatenation is the basic building block for mirrors.
State database	A state database stores information about disk about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a state database replica. The state database tracks the location and status of all known state database replicas.
State database replica	A copy of a state database. The replica ensures that the data in the database is valid.
Submirror	See RAID-0 volume.
Volume	A group of physical slices or other volumes that appear to the system as a single logical device. A volume is functionally identical to a physical disk in the view of an application or file system. In some command-line utilities, a volume is called a <i>metadevice</i> .

Examples of Using Live Upgrade to Create RAID-1 Volumes

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

Create RAID-1 Volume on Two Physical Disks

The following figure shows a new boot environment with a RAID-1 volume (mirror) that is created on two physical disks. The following command created the new boot environment and the mirror:

```
# lucreate -n second_disk -m /dev/md/dsk/d30:mirror,ufs \
-m /dev/dsk/c0t1d0s0,/dev/md/dsk/d31:attach -m /dev/dsk/c0t2d0s0,/dev/md/dsk/d32:attach \
-m -:/dev/dsk/c0t1d0s1:swap -m -:/dev/dsk/c0t2d0s1:swap
```

This command performs the following tasks:

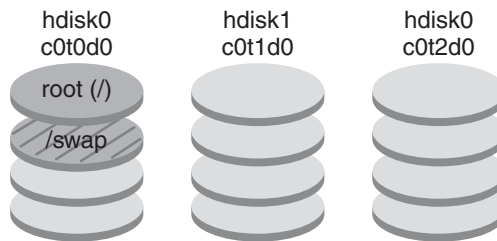
- Creates a new boot environment, `second_disk`.
- Creates a mirror `d30` and configures a UFS file system.

- Creates a single-device concatenation on slice 0 of each physical disk. The concatenations are named d31 and d32.
- Adds the two concatenations to mirror d30.
- Copies the root (/) file system to the mirror.
- Configures file systems for swap on slice 1 of each physical disk.

FIGURE 2-5 Create a Boot Environment and Create a Mirror

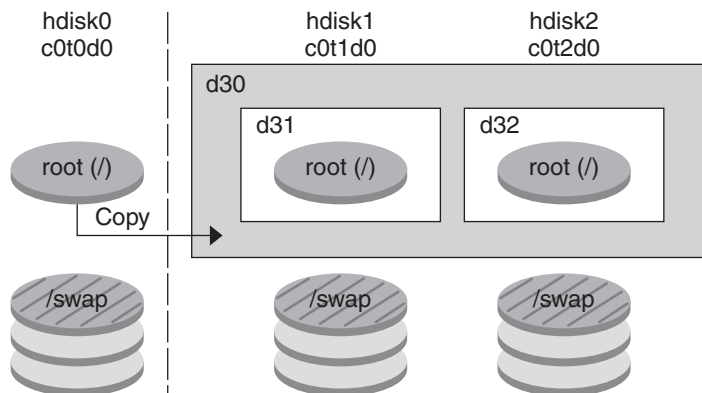
Create a New Boot Environment With a Mirror

Original system with 3 physical disks



Command: `lucreate -n second_disk -m /dev/md/dsk/d30:mirror,ufs \`
`-m /dev/dsk/c0t1d0s0,/dev/md/dsk/d31:attach \`
`-m /dev/dsk/c0t2d0s0,/dev/md/dsk/d32:attach \`
`-m -:/dev/dsk/c0t1d0s1:swap -m -:/dev/dsk/c0t2d0s1:swap`

New boot environment second_disk



d30 – RAID-1 volume (mirror)
d31 – Single-slice concatenation (submirror)
d32 – Single-slice concatenation (submirror)

Create a Boot Environment and Use the Existing Submirror

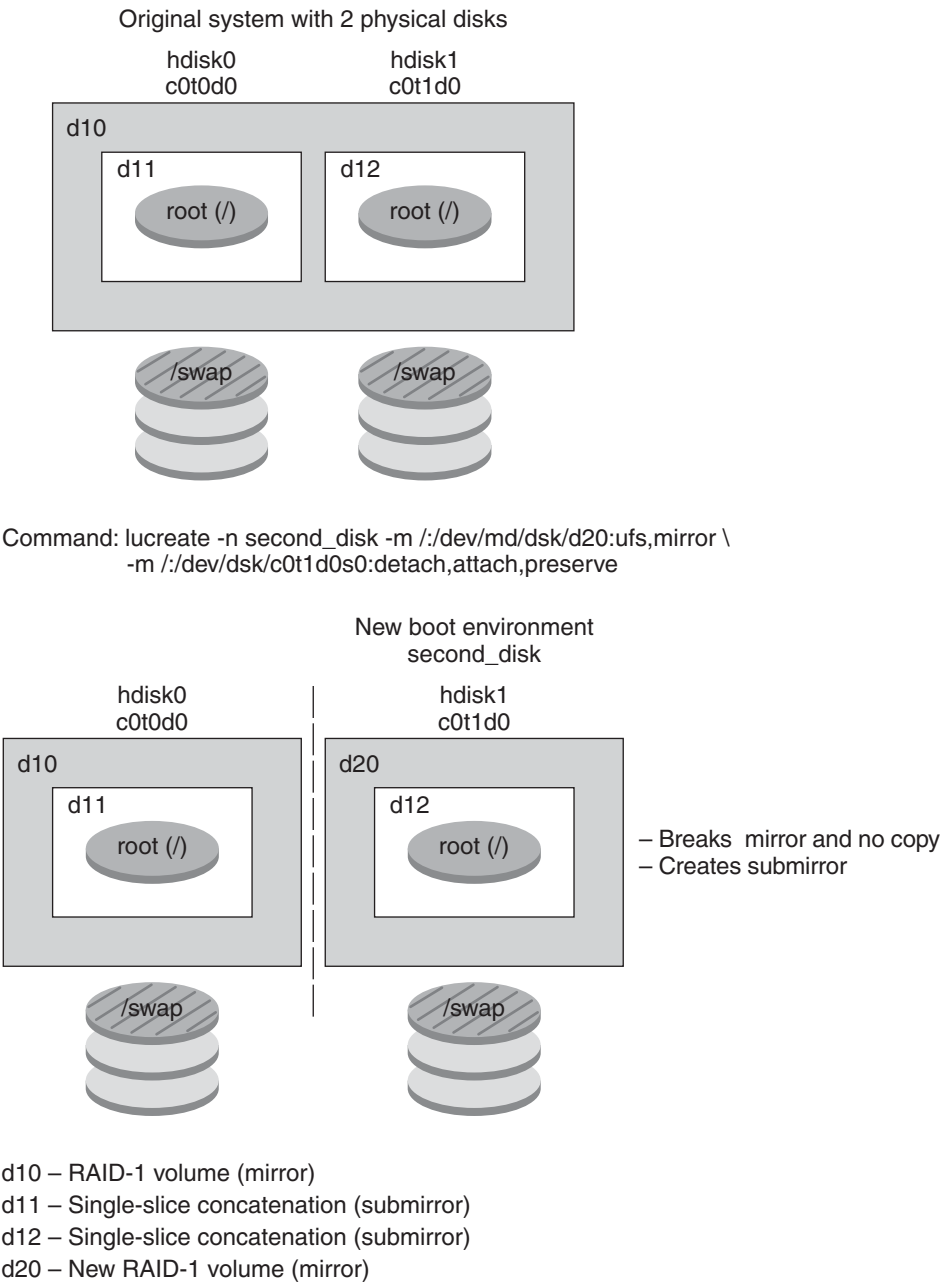
The following figure shows a new boot environment that contains a RAID-1 volume (mirror). The following command created the new boot environment and the mirror:

```
# lucreate -n second_disk -m /:/dev/md/dsk/d20:ufs,mirror \  
-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

This command performs the following tasks:

- Creates a new boot environment, `second_disk`.
- Breaks mirror `d10` and detaches concatenation `d12`.
- Preserves the contents of concatenation `d12`. File systems are not copied.
- Creates a new mirror `d20` resulting in two one-way mirrors `d10` and `d20`.
- Attaches concatenation `d12` to mirror `d20`.

FIGURE 2-6 Create a Boot Environment and Use the Existing Submirror
Create a New Boot Environment and Use the Existing Submirror



Upgrading a Boot Environment

After you have created a boot environment, you can perform an upgrade on the boot environment. As part of that upgrade, the boot environment can contain RAID-1 volumes (mirrors) for any file systems, or it can have non-global zones installed. The upgrade does not affect any files in the active boot environment. When you are ready, you activate the new boot environment, which then becomes the current boot environment.

Note – Starting with the Oracle Solaris 10 9/10 release, the upgrade process is affected by Auto Registration. See [“Auto Registration Impact for Live Upgrade” on page 36](#).

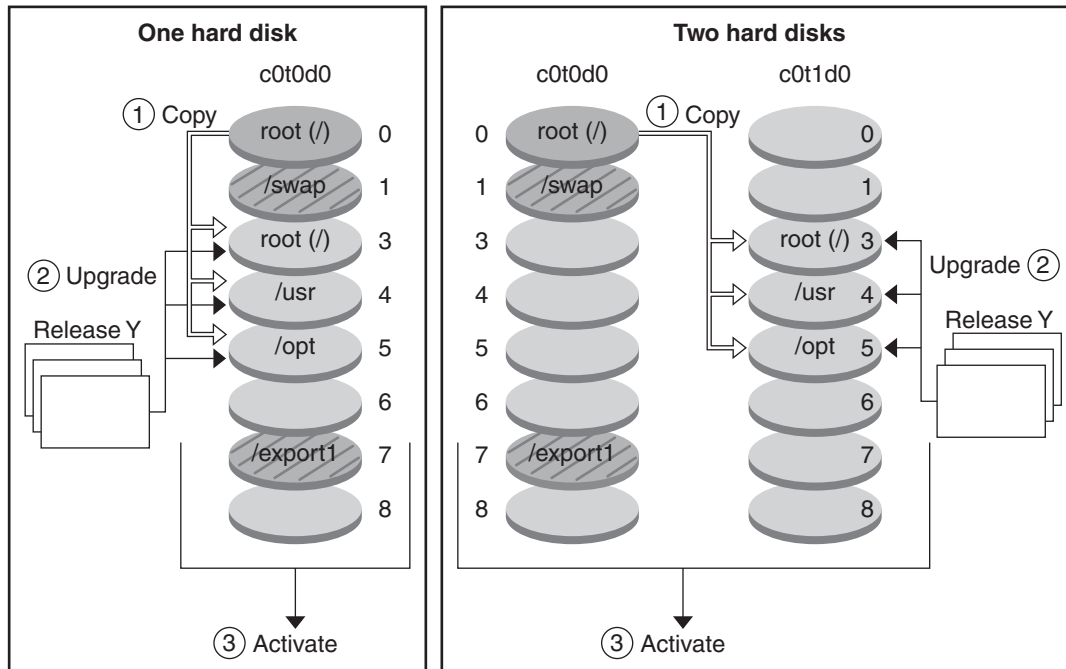
For more information, see the following resources:

- For procedures about upgrading a boot environment for UFS file systems, see [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).
- For an example of upgrading a boot environment with a RAID-1 volume file system for UFS file systems, see [“Example of Detaching and Upgrading One Side of a RAID-1 Volume \(Mirror\)” on page 156](#).
- For procedures about upgrading with non-global zones for UFS file systems, see [Chapter 8, “Upgrading the Oracle Solaris OS on a System With Non-Global Zones Installed.”](#)
- For upgrading ZFS file systems or migrating to a ZFS file system, see [Chapter 10, “Live Upgrade and ZFS \(Overview\)”](#).

The following figure shows an upgrade to an inactive boot environment.

FIGURE 2-7 Upgrading an Inactive Boot Environment

Upgrading a Boot Environment



- Current release X
Critical file system root (/)
- Inactive release Y
Critical file systems root (/) /usr /opt
- Shared file systems

- ① Single disk command:
lucreate -c solenv1 \
-m /:/dev/dsk/c0t0d0s3:ufs -m /usr:/dev/dsk/c0t0d0s4:ufs \
-m /opt:/dev/dsk/c0t0d0s5:ufs \
-n solenv2
- ① Two disks command:
lucreate -c solenv1 \
-m /:/dev/dsk/c0t1d0s3:ufs -m /usr:/dev/dsk/c0t1d0s4:ufs \
-m /opt:/dev/dsk/c0t1d0s5:ufs
-n solenv2
- ② # luupgrade -u -n solenv2 \
-s /net/installmachine/export/Solaris_10/OS_image

Rather than an upgrade, you can install a Flash Archive on a boot environment. The Flash Archive installation feature enables you to create a single reference installation of the Oracle Solaris OS on a system. This system is called the *master system*. Then, you can replicate that installation on a number of systems that are called *clone systems*. In this situation, the inactive

boot environment is a clone. When you install the flash archive on a system, the archive replaces all the files on the existing boot environment as an initial installation would.

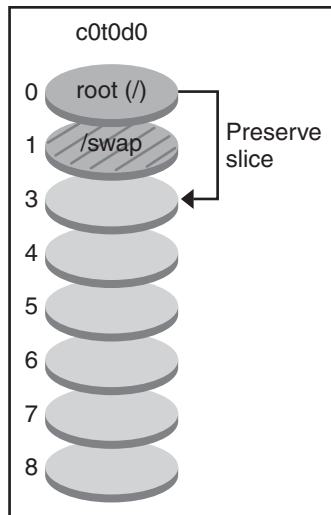
For procedures about installing a flash archive, see [“Installing Flash Archives on a Boot Environment” on page 100](#).

The following figures show an installation of a flash archive on an inactive boot environment. [Figure 2–8](#) shows a system with a single hard disk. [Figure 2–9](#) shows a system with two hard disks.

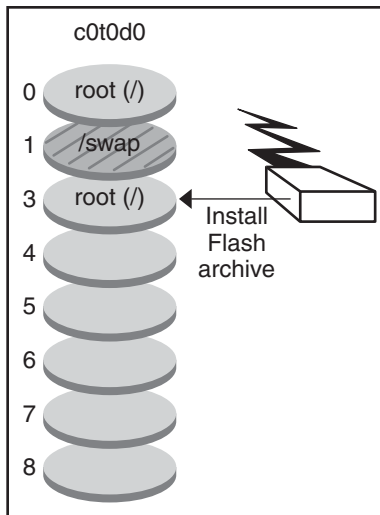
FIGURE 2–8 Installing a Flash Archive on a Single Disk




Installing a Solaris Flash Archive – Single Disk

① Create an Empty Boot Environment



② Upgrade by Installing a Flash archive



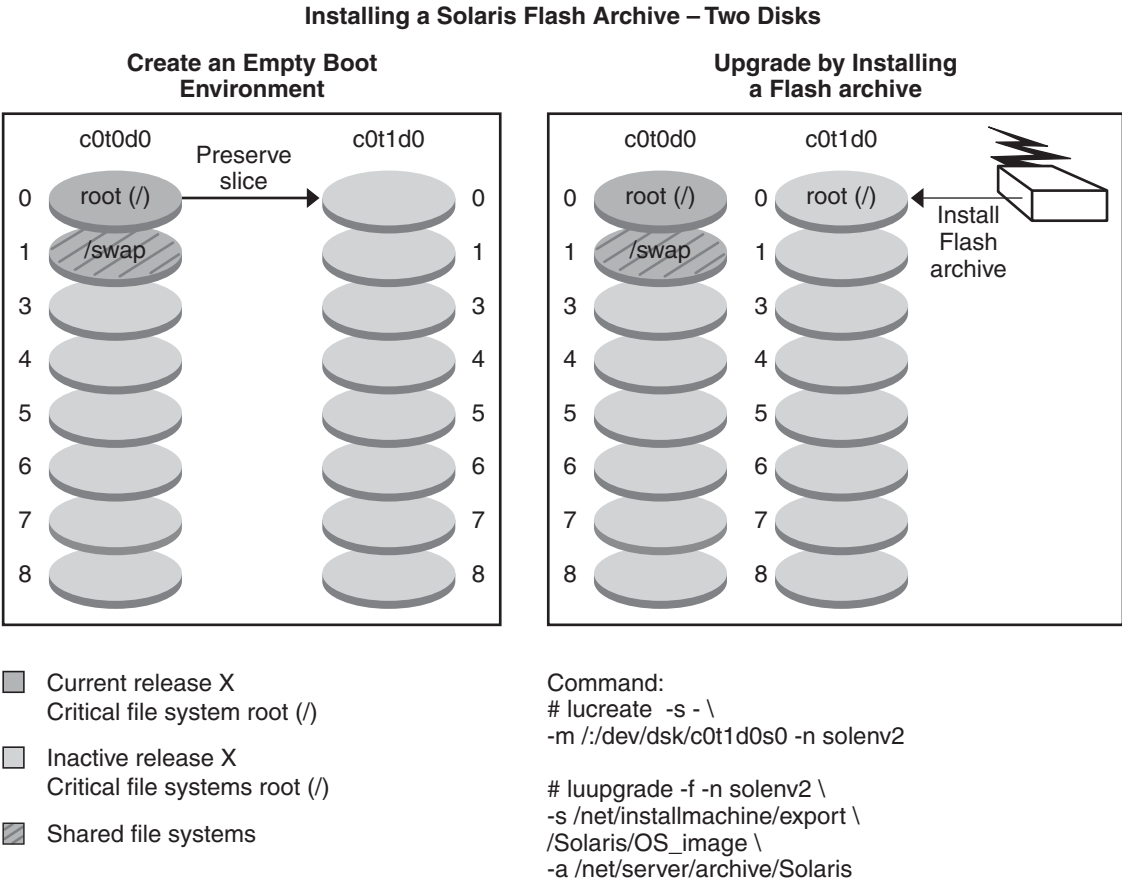
-  Current release X
Critical file systems root (/)
-  Inactive release Y
Critical file systems root (/) /usr /opt
-  Shared file systems

Command:

```
# lucreate -s - \
-m /dev/dsk/c0t0d0s3:ufs -n solenv2

# luupgrade -f -n solenv2 \
-s /net/installmachine/export \
/Solaris/OS_image \
-a /net/server/archive/Solaris
```

FIGURE 2-9 Installing a Flash Archive on Two Disks



When Does Auto Registration Affect Live Upgrade?

Auto Registration does not change Live Upgrade procedures unless you are specifically upgrading a system from a prior release to the Oracle Solaris 10 9/10 release or a later release.

Auto Registration does not change any of the following Live Upgrade procedures:

- Installing a flash archive
- Adding or removing patches or packages
- Testing a profile
- Checking package integrity

When, and only when, you are upgrading a system from a prior release to the Oracle Solaris 10 9/10 release or to a later release, you must create an Auto Registration configuration file. Then, when you upgrade that system, you must use the `-k` option in the `luupgrade -u` command and point to this configuration file.

▼ How to Provide Auto Registration Information During an Upgrade

When, and only when, you are upgrading a prior release to the Oracle Solaris 10 9/10 release or to a later release, use this procedure to provide required Auto Registration information during the upgrade.

1 Create a configuration file that contains your support credentials and, optionally, your proxy information.

This file should be formatted as a list of keyword-value pairs. Include the following keywords and values, in this format, in the file:

```
http_proxy=Proxy-Server-Host-Name
http_proxy_port=Proxy-Server-Port-Number
http_proxy_user=HTTP-Proxy-User-Name
http_proxy_pw=HTTP-Proxy-Password
oracle_user=My-Oracle-Support-User-Name
oracle_pw=My-Oracle-Support-Password
```

Note the following formatting rules:

- The passwords must be in plain, not encrypted, text.
- Keyword order does not matter.
- Keywords can be entirely omitted if you do not want to specify a value. Or, you can retain the keyword, and its value can be left blank.

Note – If you omit the support credentials, the registration will be anonymous.

- Spaces in the configuration file do not matter, unless the value you want to enter should contain a space. Only `http_proxy_user` and `http_proxy_pw` values can contain a space within the value.

- The `oracle_pw` value must not contain a space.

The following example shows a sample file.

```
http_proxy= webcache.central.example.COM
http_proxy_port=8080
http_proxy_user=webuser
http_proxy_pw=secret1
oracle_user=joe.smith@example.com
oracle_pw=csdfl2442IJS
```

- 2 Save the file.
- 3 Run the `luupgrade -u -k /path/filename` command, including any of the other standard `luupgrade` command options as needed for that particular upgrade.

▼ How to Disable Auto Registration During an Upgrade

- 1 Create a configuration file or revise the content of the existing configuration file you created so that the file contains only the following line:

```
autoreg=disable
```

- 2 Save the file.
- 3 Run the `luupgrade -u -k /path/filename` command, including any of the other standard `luupgrade` command options as needed for that particular upgrade.
- 4 (Optional) When the Live Upgrade has completed and the system reboots, verify that the Auto Registration feature is disabled as follows.

```
# /opt/ocm/ccr/bin/emCCR status
Oracle Configuration Manager - Release: 10.3.6.0.1 - Production
Copyright (c) 2005, 2011, Oracle and/or its affiliates. All rights reserved.
-----
Log Directory          /opt/ocm/config_home/ccr/log
Collector Mode         Disconnected
```

Activating a Boot Environment

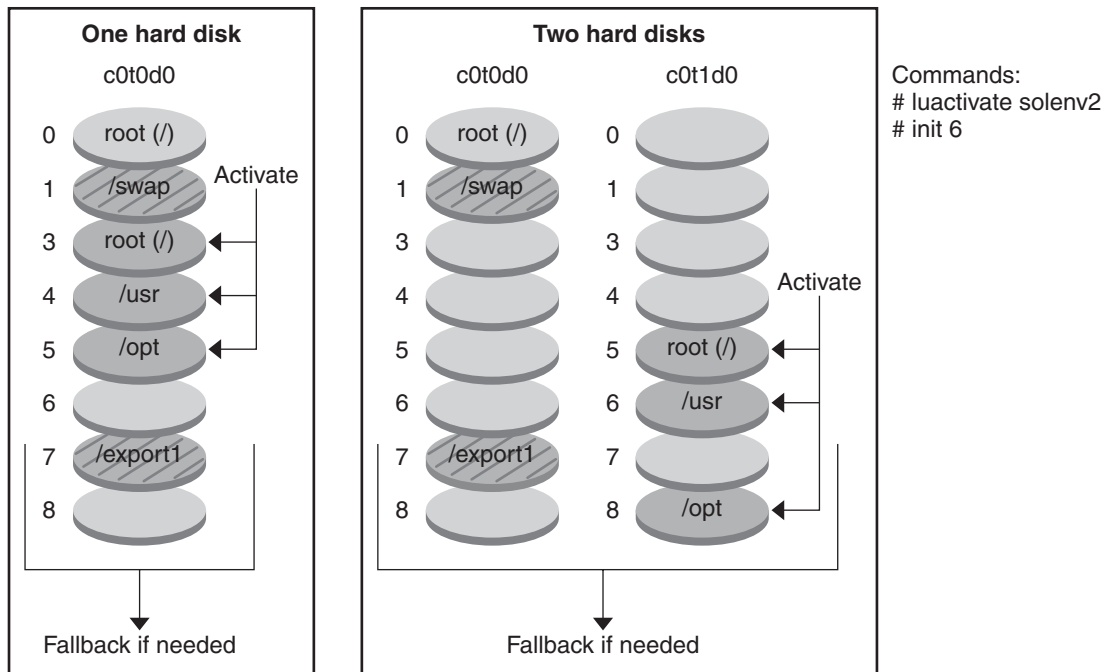
When you are ready to switch and make the new boot environment active, you can easily activate the new boot environment and reboot. Files are synchronized between boot environments the first time that you boot a newly created boot environment. “Synchronize” means that certain system files and directories are copied from the last-active boot environment to the boot environment being booted. When you reboot the system, the configuration that you installed on the new boot environment is active. The original boot environment then becomes an inactive boot environment.

For procedures about activating a boot environment, see [“Activating a Boot Environment” on page 104](#). For information about synchronizing the active and inactive boot environment, see [“Synchronizing Files Between Boot Environments” on page 54](#).

The following figure shows a switch after a reboot from an inactive to an active boot environment.

FIGURE 2-10 Activating an Inactive Boot Environment

Activating a Boot Environment



- Current release Y
Critical file systems root (/) /usr /opt
- Inactive release X
Critical file systems root (/)
- Shared file systems

Falling Back to the Original Boot Environment

If a failure occurs, you can quickly fall back to the original boot environment with an activation and reboot. The use of fallback takes only the time to reboot the system, which is much quicker

than backing up and restoring the original. The new boot environment that failed to boot is preserved. The failure can then be analyzed. You can fall back only to the boot environment that was used by `luactivate` to activate the new boot environment.

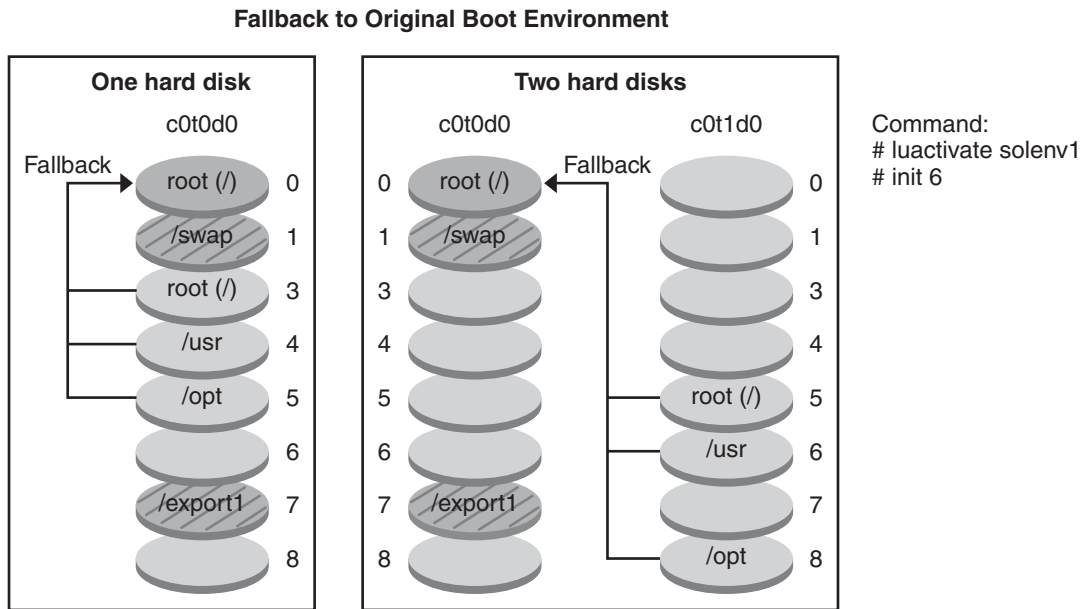
The following table describes the ways you can fall back to the previous boot environment.

Problem	Action
The new boot environment boots successfully, but you are not happy with the results.	Run the <code>luactivate</code> command with the name of the previous boot environment and reboot. x86 only – Starting with the Solaris 10 1/06 release , you can fall back by selecting the original boot environment that is found on the GRUB menu. The original boot environment and the new boot environment must be based on the GRUB software. Booting from the GRUB menu does not synchronize files between the old and new boot environments. For more information about synchronizing files, see “Forcing a Synchronization Between Boot Environments” on page 55 .
The new boot environment does not boot.	Boot the fallback boot environment in single-user mode, run the <code>luactivate</code> command, and reboot.
You cannot boot in single-user mode.	Perform one of the following: <ul style="list-style-type: none">■ Boot from DVD or CD media or a net installation image■ Mount the root (/) file system on the fallback boot environment■ Run the <code>luactivate</code> command and reboot

For procedures to fall back, see [Chapter 6, “Failure Recovery: Falling Back to the Original Boot Environment \(Tasks\)”](#).

The following figure shows the switch that is made when you reboot to fallback.

FIGURE 2-11 Fallback to the Original Boot Environment



- Current release X
Critical file system root (/)
- Inactive release X
Critical file systems root (/)
- ▨ Shared file systems

Maintaining a Boot Environment

You can also do various maintenance activities such as checking status, renaming, or deleting a boot environment. For maintenance procedures, see [Chapter 7, “Maintaining Live Upgrade Boot Environments \(Tasks\)”](#).

Live Upgrade (Planning)

This chapter provides guidelines and requirements for review before installing and using Live Upgrade, a feature of Oracle Solaris. You also should review general information about upgrading in “Upgrade Planning” in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*.

Note – This chapter describes Live Upgrade for UFS file systems. For planning information for migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see Chapter 11, “Live Upgrade for ZFS (Planning).”

This chapter contains the following sections:

- “Live Upgrade Requirements” on page 43
- “Upgrading a System With Packages or Patches” on page 47
- “Guidelines for Creating File Systems With the `lucreate` Command” on page 48
- “Guidelines for Selecting Slices for File Systems” on page 49
- “Customizing a New Boot Environment’s Content” on page 53
- “Synchronizing Files Between Boot Environments” on page 54

Live Upgrade Requirements

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

Live Upgrade System Requirements

Live Upgrade is included in the Oracle Solaris software. You need to install the Live Upgrade packages on your current OS. The release of the Live Upgrade packages must match the release of the OS you are upgrading to. For example, if your current OS is the Solaris 9 release and you want to upgrade to the Oracle Solaris 10 8/11 release, you need to install the Live Upgrade packages from the Oracle Solaris 10 8/11 release.

The following table lists releases that are supported by Live Upgrade.

TABLE 3-1 Supported Oracle Solaris Releases

Your Current Release	Compatible Upgrade Release
Solaris 8 OS	Solaris 8, Solaris 9, or any Oracle Solaris 10 release
Solaris 9 OS	Solaris 9 or any Oracle Solaris 10 release
Oracle Solaris 10 OS	Any Oracle Solaris 10 release

Installing Live Upgrade

You can install the Live Upgrade packages by using either of the following methods:

- The `pkgadd` command. The Live Upgrade packages are `SUNWlucfg`, `SUNWlur`, and `SUNWluu`, and these packages must be installed in that order.
- An installer on the Oracle Solaris Operating System DVD, the Oracle Solaris Software - 2 CD, or a network installation image.

Note the following information about patches that might need to be installed for the correct operation of Live Upgrade.

Correct operation of Live Upgrade requires that a limited set of patch revisions be installed for a particular OS version. Before installing or running Live Upgrade, you are required to install these patches.

x86 only – If this set of patches is not installed, Live Upgrade fails and you might see the following error message. If you don't see the following error message, necessary patches still might not be installed. Always verify that all patches listed on the My Oracle Support knowledge document have been installed before attempting to install Live Upgrade.

```
ERROR: Cannot find or is not executable:
</sbin/biosdev>.
ERROR: One or more patches required
by Live Upgrade has not been installed.
```

Ensure that you have the most recently updated patch list by consulting <http://support.oracle.com> (My Oracle Support). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

The patches listed in knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support are subject to change at any time. These patches potentially fix defects in Live Upgrade, as well as fix defects in components that Live

Upgrade depends on. If you experience any difficulties with Live Upgrade, please check and make sure that you have the latest Live Upgrade patches installed.

If you are running the Solaris 8 or Solaris 9 OS, you might not be able to run the Live Upgrade installer. These releases do not contain the set of patches needed to run the Java 2 runtime environment. You must have the recommended patch cluster for the Java 2 runtime environment recommended to run the Live Upgrade installer and install the packages. To install the Live Upgrade packages, use the `pkgadd` command. Or install, for the Java 2 runtime environment, the recommended patch cluster. The patch cluster is available on <http://support.oracle.com> (My Oracle Support).

For instructions about installing the Live Upgrade software, see “Installing Live Upgrade” on page 60.

Required Packages

If you have problems with Live Upgrade, you might be missing packages. Table 3–2 lists the packages that each OS release requires to use Live Upgrade.

For the Oracle Solaris 10 release:

- The following software groups contain all the required Live Upgrade packages:
 - Entire Oracle Solaris Software Group Plus OEM Support
 - Entire Oracle Solaris Software Group
 - Developer Oracle Solaris Software Group
 - End User Oracle Solaris Software Group
- The following software groups might not include all the packages required to use Live Upgrade:
 - Core System Support Software Group
 - Reduced Network Support Software Group

For information about software groups, see “Disk Space Recommendations for Software Groups” in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*.

TABLE 3–2 Required Packages for Live Upgrade

Solaris 8 Release	Solaris 9 Release	Oracle Solaris 10 Release
SUNWadmap	SUNWadmap	SUNWadmap
SUNWadmc	SUNWadmc	SUNWadmlib-sysid
SUNWlibC	SUNWadmfw	SUNWadmr
SUNWbzip	SUNWlibC	SUNWlibC
SUNWgzip	SUNWgzip	For Solaris 10 3/05 only: SUNWgzip

TABLE 3-2 Required Packages for Live Upgrade (Continued)

Solaris 8 Release	Solaris 9 Release	Oracle Solaris 10 Release
SUNWj2rt ¹	SUNWj2rt	SUNWj5rt ²

¹ The SUNWj2rt package is needed only under the following conditions:

- When you run the Live Upgrade installer to add Live Upgrade packages
- When you upgrade and use CD media

² The SUNWj5rt package is needed only under the following conditions:

- When you run the Live Upgrade installer to add Live Upgrade packages
- When you upgrade and use CD media

To check for packages on your system, type the following command.

```
% pkginfo package-name
```

Live Upgrade Disk Space Requirements

For information about disk space requirements for an upgrade, see [Chapter 3, “System Requirements, Guidelines, and Upgrade Information,” in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*](#).

To estimate the file system size that is needed to create a boot environment, start the creation of a new boot environment and note the size is calculated. You can then quit the process.

The disk on the new boot environment must be able to serve as a boot device. Some systems restrict which disks can serve as a boot device. Refer to your system's documentation to determine whether any boot restrictions apply.

The disk might need to be prepared before you create the new boot environment. Check that the disk is formatted properly as follows:

- Identify slices large enough to hold the file systems to be copied.
- Identify file systems that contain directories that you want to share between boot environments rather than copy. If you want a directory to be shared, you need to create a new boot environment with the directory put on its own slice. The directory is then a file system and can be shared with future boot environments. For more information about creating separate file systems for sharing, see [“Guidelines for Selecting Slices for Shareable File Systems” on page 52](#).

Live Upgrade Requirements If Creating RAID-1 Volumes (Mirrors)

Live Upgrade uses Solaris Volume Manager technology to create a boot environment that can contain file systems that are RAID-1 volumes (mirrors). Live Upgrade does not implement the full functionality of Solaris Volume Manager, but does require the following components of Solaris Volume Manager.

TABLE 3-3 Required Components for Live Upgrade and RAID-1 Volumes

Requirement	Description	For More Information
You must create at least one state database and at least three state database replicas.	A state database stores information about disk about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a state database replica. When a state database is copied, the replica protects against data loss from single points of failure.	For information about creating a state database, see Chapter 6, “State Database (Overview),” in <i>Solaris Volume Manager Administration Guide</i> .
Live Upgrade supports only a RAID-1 volume (mirror) with single-slice concatenations on the root (/) file system.	<p>A concatenation is a RAID-0 volume. If slices are concatenated, the data is written to the first available slice until that slice is full. When that slice is full, the data is written to the next slice, serially. A concatenation provides no data redundancy unless it is contained in a RAID-1 volume</p> <p>A RAID-1 volume can be comprised of a maximum of three concatenations.</p>	For guidelines about creating mirrored file systems, see “ Guidelines for Selecting Slices for Mirrored File Systems ” on page 50.

Upgrading a System With Packages or Patches

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



Caution – When upgrading and adding and removing packages or patches, Live Upgrade requires packages or patches that comply with the SVR4 advanced packaging guidelines. While Oracle packages conform to these guidelines, Oracle cannot guarantee the conformance of packages from third-party vendors. If a package violates these guidelines, the package can cause the package-addition software during an upgrade to fail or alter the active boot environment.

For more information about packaging requirements, see [Appendix C, “Additional SVR4 Packaging Requirements \(Reference\).”](#)

Type of Installation	Description	For More Information
Adding patches to a boot environment	Create a new boot environment and use the <code>luupgrade</code> command with the <code>-t</code> option.	“Adding Patches to or Removing Patches From a Network Installation Image on a Boot Environment” on page 91
Adding packages to a boot environment	Use the <code>luupgrade</code> command with the <code>-p</code> option.	“Adding Packages to or Removing Packages From a Network Installation Image on a Boot Environment” on page 90
Using Live Upgrade to install a flash archive	An archive contains a complete copy of a boot environment with new packages and patches already included. This copy can be installed on multiple systems.	<ul style="list-style-type: none">■ For details about how to create a flash archive, see Chapter 3, “Creating Flash Archives (Tasks)” in <i>Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation)</i>■ For information about using Live Upgrade to install a flash archive, see “Installing Flash Archives on a Boot Environment” on page 100

For information about upgrading and patching limitations, see [“Upgrading and Patching Limitations” in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*](#).

Guidelines for Creating File Systems With the `lucreate` Command

The `lucreate -m` option specifies the specific file systems and the number of file systems to be created in the new boot environment. You must specify the exact number of file systems you want to create by repeating this option. When using the `-m` option to create file systems, follow these guidelines:

- You must specify one `-m` option for the root (`/`) file system for the new boot environment. If you run `lucreate` without the `-m` option, the Configuration menu is displayed. The Configuration menu enables you to customize the new boot environment by redirecting files onto new mount points.
- Any critical file systems that exist in the current boot environment and that are not specified in a `-m` option are merged into the next highest level file system created.
- Only the file systems that are specified by the `-m` option are created on the new boot environment. To create the same number of files systems that is on your current system, you must specify one `-m` option for each file system to be created.

For example, a single use of the `-m` option specifies where to put all the file systems. You merge all the file systems from the original boot environment into the one file system that is specified by the `-m` option. If you specify the `-m` option twice, you create two file systems. If you have file systems for root (/), /opt, and /var, you would use one `-m` option for each file system on the new boot environment.

- Do not duplicate a mount point. For example, you cannot have two root (/) file systems.

Guidelines for Selecting Slices for File Systems

When you create file systems for a boot environment, the rules are identical to the rules for creating file systems for the Oracle Solaris OS. Live Upgrade cannot prevent you from creating invalid configurations for critical file systems. For example, you could type a `lucreate` command that would create separate file systems for root (/) and /kernel which is an invalid division of the root (/) file system.

Do not overlap slices when reslicing disks. If this condition exists, the new boot environment appears to have been created, but when activated, the boot environment does not boot. The overlapping file systems might be corrupted.

For Live Upgrade to work properly, the `vfstab` file on the active boot environment must have valid contents and must have an entry for the root (/) file system at a minimum.

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

When you create an inactive boot environment, you need to identify a slice where the root (/) file system is to be copied. Use the following guidelines when you select a slice for the root (/) file system.

- Must be a slice from which the system can boot.
- Must meet the recommended minimum size.
- Can be on different physical disks or the same disk as the active root (/) file system.
- Can be a Veritas Volume Manager volume (VxVM). If VxVM volumes are configured on your current system, the `lucreate` command can create a new boot environment. When the data is copied to the new boot environment, the Veritas file system configuration is lost and a UFS file system is created on the new boot environment.

Guidelines for Selecting Slices for Mirrored File Systems

You can create a new boot environment that contains any combination of physical disk slices, Solaris Volume Manager volumes, or Veritas Volume Manager volumes. Critical file systems that are copied to the new boot environment can be of the following types:

- A physical slice.
- A single-slice concatenation that is included in a RAID-1 volume (mirror). The slice that contains the root (/) file system can be a RAID-1 volume.
- A single-slice concatenation that is included in a RAID-0 volume. The slice that contains the root (/) file system can be a RAID-0 volume.

When you create a new boot environment, the `lucreate -m` command recognizes the following three types of devices:

- A physical slice in the form of `/dev/dsk/cwtxdysz`
- A Solaris Volume Manager volume in the form of `/dev/md/dsk/dnum`
- A Veritas Volume Manager volume in the form of `/dev/vx/dsk/volume_name`. If VxVM volumes are configured on your current system, the `lucreate` command can create a new boot environment. When the data is copied to the new boot environment, the Veritas file system configuration is lost and a UFS file system is created on the new boot environment.

Note – If you have problems upgrading with Veritas VxVM, see [“How to Upgrade When Running Veritas VxVm” on page 225](#).

General Guidelines When Creating RAID-1 Volumes (Mirrored) File Systems

Use the following guidelines to check whether a RAID-1 volume is busy or resyncing, or if volumes contain file systems that are in use by a Live Upgrade boot environment.

For volume naming guidelines, see [“RAID Volume Name Requirements and Guidelines” in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*](#).

Checking the Status of Volumes

If a mirror or submirror needs maintenance or is busy, components cannot be detached. You should use the `metastat` command before creating a new boot environment and using the `detach` keyword. The `metastat` command checks whether the mirror is in the process of resynchronization or is in use. For information, see the [`metastat\(1M\)`](#) man page.

Detaching Volumes and Resynchronizing Mirrors

If you use the `detach` keyword to detach a submirror, `lucreate` checks if a device is currently resyncing. If the device is resyncing, you cannot detach the submirror and you see an error message.

Resynchronization is the process of copying data from one submirror to another submirror after the following problems:

- Submirror failures
- System crashes
- A submirror has been taken offline and brought back online
- The addition of a new submirror

For more information about resynchronization, see [“RAID-1 Volume \(Mirror\) Resynchronization”](#) in *Solaris Volume Manager Administration Guide*.

Using Solaris Volume Manager Commands

Use the `lucreate` command rather than Solaris Volume Manager commands to manipulate volumes on inactive boot environments. The Solaris Volume Manager software has no knowledge of boot environments, whereas the `lucreate` command contains checks that prevent you from inadvertently destroying a boot environment. For example, `lucreate` prevents you from overwriting or deleting a Solaris Volume Manager volume.

However, if you have already used Solaris Volume Manager software to create complex Solaris Volume Manager concatenations, stripes, and mirrors, you must use Solaris Volume Manager software to manipulate them. Live Upgrade is aware of these components and supports their use. Before using Solaris Volume Manager commands that can create, modify, or destroy volume components, use the `lustatus` or `luflist` commands. These commands can determine which Solaris Volume Manager volumes contain file systems that are in use by a Live Upgrade boot environment.

Guidelines for Selecting a Slice for a Swap Volume

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

Configuring Swap for the New Boot Environment

You can configure a swap slice in three ways by using the `lucreate` command with the `-m` option:

- If you do not specify a swap slice, the swap slices belonging to the current boot environment are configured for the new boot environment.
- If you specify one or more swap slices, these slices are the only swap slices that are used by the new boot environment. The two boot environments do not share any swap slices.

- You can specify to both share a swap slice and add a new slice for swap.

The following examples show the three ways of configuring swap. The current boot environment is configured with the root (/) file system on `c0t0d0s0`. The swap file system is on `c0t0d0s1`.

- In the following example, no swap slice is specified. The new boot environment contains the root (/) file system on `c0t1d0s0`. Swap is shared between the current and new boot environment on `c0t0d0s1`.

```
# lucreate -n be2 -m /:/dev/dsk/c0t1d0s0:ufs
```

- In the following example, a swap slice is specified. The new boot environment contains the root (/) file system on `c0t1d0s0`. A new swap file system is created on `c0t1d0s1`. No swap slice is shared between the current and new boot environment.

```
# lucreate -n be2 -m /:/dev/dsk/c0t1d0s0:ufs -m -:/dev/dsk/c0t1d0s1:swap
```

- In the following example, a swap slice is added and another swap slice is shared between the two boot environments. The new boot environment contains the root (/) file system on `c0t1d0s0`. A new swap slice is created on `c0t1d0s1`. The swap slice on `c0t0d0s1` is shared between the current and new boot environment.

```
# lucreate -n be2 -m /:/dev/dsk/c0t1d0s0:ufs -m -:shared:swap \  
-m -:/dev/dsk/c0t1d0s1:swap
```

Failed Boot Environment Creation If Swap Is in Use

A boot environment creation fails if the swap slice is being used by any boot environment except for the current boot environment. If the boot environment was created using the `-s` option, the alternate-source boot environment can use the swap slice, but not any other boot environment.

Guidelines for Selecting Slices for Shareable File Systems

Live Upgrade copies the entire contents of a slice to the designated new boot environment slice. You might want some large file systems on that slice to be shared between boot environments rather than copied to conserve space and copying time. File systems that are critical to the OS such as root (/) and /var must be copied. File systems such as /home are not critical file systems and could be shared between boot environments. Shareable file systems must be user-defined file systems and on separate swap slices on both the active and new boot environments. You can reconfigure the disk several ways, depending your needs.

Reconfiguring a disk	Examples	For More Information
You can reslice the disk before creating the new boot environment and put the shareable file system on its own slice.	For example, if the root (/) file system, /var, and /home are on the same slice, reconfigure the disk and put /home on its own slice. When you create any new boot environments, /home is shared with the new boot environment by default.	format(1M)
If you want to share a directory, the directory must be split off to its own slice. The directory is then a file system that can be shared with another boot environment. You can use the <code>lucreate</code> command with the <code>-m</code> option to create a new boot environment and split a directory off to its own slice. But, the new file system cannot yet be shared with the original boot environment. You need to run the <code>lucreate</code> command with the <code>-m</code> option again to create another boot environment. The two new boot environments can then share the directory.	For example, if you wanted to upgrade from the Solaris 9 release to the Oracle Solaris 10 8/11 release and share /home, you could run the <code>lucreate</code> command with the <code>-m</code> option. You could create a Solaris 9 release with /home as a separate file system on its own slice. Then run the <code>lucreate</code> command with the <code>-m</code> option again to duplicate that boot environment. This third boot environment can then be upgraded to the Oracle Solaris 10 8/11 release. /home is shared between the Solaris 9 and Oracle Solaris 10 8/11 releases.	For a description of shareable and critical file systems, see “File System Types” on page 22 .

Customizing a New Boot Environment's Content

When you create a new boot environment, some directories and files can be excluded from a copy to the new boot environment. If you have excluded a directory, you can also reinstate specified subdirectories or files under the excluded directory. These subdirectories or files that have been restored are then copied to the new boot environment. For example, you could exclude from the copy all files and directories in `/etc/mail` but include all files and directories in `/etc/mail/staff`. For example, the following command copies the `staff` subdirectory to the new boot environment.

```
# lucreate -n second_disk -x /etc/mail -y /etc/mail/staff
```



Caution – Use the file-exclusion options with caution. Do not remove files or directories that are required by the system.

For examples of customizing the directories and files when creating a boot environment, see [“Creating a Boot Environment and Customizing the Content” on page 83](#).

Synchronizing Files Between Boot Environments

When you are ready to switch and make the new boot environment active, you quickly activate the new boot environment and reboot. Files are synchronized between boot environments the first time that you boot a newly created boot environment. “Synchronize” means that certain critical system files and directories might be copied from the last-active boot environment to the boot environment being booted. Those files and directories that have changed are copied.

Adding Files to the `/etc/lu/syncList` File

Live Upgrade checks for critical files that have changed. If the content of these files is not the same in both boot environments, they are copied from the active boot environment to the new boot environment. Synchronizing is meant for critical files such as `/etc/passwd` or `/etc/group` files that might have changed since the new boot environment was created.

The `/etc/lu/syncList` file contains a list of directories and files that are synchronized. In some instances, you might want to copy other files from the active boot environment to the new boot environment. You can add directories and files to `/etc/lu/syncList` if necessary.

Adding files not listed in the `/etc/lu/syncList` could cause a system to become unbootable. The synchronization process only copies files and creates directories. The process does not remove files and directories.

The following example of the `/etc/lu/syncList` file shows the standard directories and files that are synchronized for this system.

<code>/var/mail</code>	OVERWRITE
<code>/var/spool/mqueue</code>	OVERWRITE
<code>/var/spool/cron/crontabs</code>	OVERWRITE
<code>/var/dhcp</code>	OVERWRITE
<code>/etc/passwd</code>	OVERWRITE
<code>/etc/shadow</code>	OVERWRITE
<code>/etc/opasswd</code>	OVERWRITE
<code>/etc/oshadow</code>	OVERWRITE
<code>/etc/group</code>	OVERWRITE
<code>/etc/pwhist</code>	OVERWRITE
<code>/etc/default/passwd</code>	OVERWRITE
<code>/etc/dfs</code>	OVERWRITE
<code>/var/log/syslog</code>	APPEND
<code>/var/adm/messages</code>	APPEND

The following example shows directories and files that might be appropriate to add to the `syncList` file.

<code>/var/yp</code>	OVERWRITE
<code>/etc/mail</code>	OVERWRITE
<code>/etc/resolv.conf</code>	OVERWRITE
<code>/etc/domainname</code>	OVERWRITE

The `sync` list file entries can be files or directories. The second field indicates the method of updating that occurs on the activation of the boot environment. You can choose from three methods to update files:

- **OVERWRITE** – The contents of the active boot environment's file overwrite the contents of the new boot environment file. **OVERWRITE** is the default action if no action is specified in the second field. If the entry is a directory, all subdirectories are copied. All files are overwritten. The new boot environment file has the same date, mode, and ownership as the same file on the previous boot environment.
- **APPEND** – The contents of the active boot environment's file are added to the end of the new boot environment's file. This addition might lead to duplicate entries in the file. Directories cannot be listed as **APPEND**. The new boot environment file has the same date, mode, and ownership as the same file on the previous boot environment.
- **PREPEND** – The contents of the active boot environment's file are added to the beginning of the new boot environment's file. This addition might lead to duplicate entries in the file. Directories can not be listed as **PREPEND**. The new boot environment file has the same date, mode, and ownership as the same file on the previous boot environment.

Forcing a Synchronization Between Boot Environments

The first time you boot from a newly created boot environment, Live Upgrade synchronizes the new boot environment with the boot environment that was last active. After this initial boot and synchronization, Live Upgrade does not perform a synchronization unless requested. To force a synchronization, you use the `luactivate` command with the `-s` option.

You might want to force a synchronization if you are maintaining multiple versions of the Oracle Solaris OS. You might want changes in files such as `email` or `passwd/group` to be in the boot environment you are activating to. If you force a synchronization, Live Upgrade checks for conflicts between files that are subject to synchronization. When the new boot environment is booted and a conflict is detected, a warning is issued and the files are not synchronized. Activation can be completed successfully, despite such a conflict. A conflict can occur if you make changes to the same file on both the new boot environment and the active boot environment. For example, if you make changes to the `/etc/passwd` file on the original boot environment and then make other changes to `/etc/passwd` file on the new boot environment, the synchronization process cannot choose which file to copy for the synchronization.



Caution – Use this option with great care, because you might not be aware of or in control of changes that might have occurred in the last active boot environment. For example, if you were running Oracle Solaris 10 8/11 software on your current boot environment and booted back to a Solaris 9 release with a forced synchronization, files could be changed on the Solaris 9 release. Because files are dependent on the release of the OS, the boot to the Solaris 9 release could fail because the Oracle Solaris 10 8/11 files might not be compatible with the Solaris 9 files.

Booting Multiple Boot Environments

If your system has more than one OS installed on the system, you can boot from these boot environments for both SPARC and x86 platforms. The boot environments available for booting include Live Upgrade inactive boot environments.

- **Starting with the Solaris 10 10/08 release** for a SPARC based system, you can boot a ZFS root file system in a ZFS pool. For ZFS root pools, you can list the available boot environments with the boot command with the -L option. You can then choose a boot environment and use the OBP boot command with the -Z option to boot that boot environment. The -Z option is an alternative for the luactivate command that is also used to boot a new boot environment for a ZFS root pool. The luactivate command is the preferred method of switching boot environments. For a UFS file system, you continue to use the OpenBoot PROM OBP as the primary administrative interface, with boot options selected by using OBP commands.
- **Starting with the Solaris 10 1/06 release** for x86 based systems, a GRUB boot menu provides the interface for booting between different boot environments. Starting with the Solaris 10 10/08 release, this menu lists ZFS boot environments that are available for booting. If the default boot environment is a ZFS file system and the GRUB menu is displayed, you can let the default boot environment boot or choose another boot environment to boot. The GRUB menu is an alternative to using the luactivate command that is also used to boot a new boot environment for a ZFS root pool. The luactivate command is the preferred method of switching boot environments.

On both SPARC and x86 based systems, each ZFS root pool has a dataset designated as the default root file system. If for SPARC you type the boot command or for x86 you take the default from the GRUB menu, then this default root file system is booted.

Note – If the GRUB menu has been explicitly modified to designate a default menu item other than the one set by Live Upgrade, then selecting that default menu entry might not result in the booting of the pool's default root file system.

For more information about booting and modifying the GRUB boot menu, see the following references.

Task	Information
To activate a boot environment with the GRUB menu	“x86: How to Activate a Boot Environment With the GRUB Menu” on page 109
To fall back to the original boot environment with a GRUB menu	“x86: To Fall Back Despite Successful New Boot Environment Activation With the GRUB Menu” on page 114
For SPARC and x86 information and step-by-step procedures for booting and modifying boot behavior	<i>Oracle Solaris Administration: Basic Administration</i> <ul style="list-style-type: none">■ Chapter 8, “Introduction to Shutting Down and Booting a System,” in <i>Oracle Solaris Administration: Basic Administration</i>■ Chapter 9, “Shutting Down and Booting a System (Overview),” in <i>Oracle Solaris Administration: Basic Administration</i>■ Chapter 12, “Booting an Oracle Solaris System (Tasks),” in <i>Oracle Solaris Administration: Basic Administration</i>■ Chapter 11, “Modifying Oracle Solaris Boot Behavior (Tasks),” in <i>Oracle Solaris Administration: Basic Administration</i>■ Chapter 13, “Managing the Oracle Solaris Boot Archives (Tasks),” in <i>Oracle Solaris Administration: Basic Administration</i>
For an overview and step-by-step procedures for booting ZFS boot environments	“Booting From a ZFS Root File System” in <i>Oracle Solaris ZFS Administration Guide</i>

Live Upgrade Character User Interface

Oracle no longer recommends use of the `lu` command, which displays a character user interface (CUI). The underlying command sequence for the CUI, typically the `lucreate`, `luupgrade`, and `luactivate` commands, is straightforward to use. Procedures for these commands are provided in the following chapters.

Using Live Upgrade to Create a Boot Environment (Tasks)

This chapter explains how to install Live Upgrade packages and patches and how to create a boot environment.

Note – This chapter describes Live Upgrade for UFS file systems. For procedures for migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see [Chapter 12](#), “Creating a Boot Environment for ZFS Root Pools.”

This chapter contains the following sections:

- “Installing Live Upgrade and Creating Boot Environments (Task Map)” on page 59
- “Installing Live Upgrade” on page 60
- “Creating a New Boot Environment” on page 64

Installing Live Upgrade and Creating Boot Environments (Task Map)

TABLE 4-1 Task Map: Using Live Upgrade

Task	Description	For Instructions
Install Live Upgrade packages	Install packages on your OS	“Installing Live Upgrade” on page 60
Install patches on your system	Live Upgrade requires a limited set of patch revisions	“Installing Patches Needed by Live Upgrade” on page 62
Create a boot environment	Copy and reconfigure file systems to an inactive boot environment	“Creating a New Boot Environment” on page 64

Installing Live Upgrade

Before running Live Upgrade, you must install the latest Live Upgrade packages from installation media and install the patches listed in the My Oracle Support knowledge document 1004881.1 – Live Upgrade Software Patch Requirements (formerly 206844). You need to install the Live Upgrade packages on your current OS and remove old packages. The release of the Live Upgrade packages must match the release of the OS you are upgrading to. For example, if your current OS is the Solaris 9 release and you want to upgrade to the Oracle Solaris 10 8/11 release, you need to install the Live Upgrade packages from the Oracle Solaris 10 8/11 release. The patches listed in My Oracle Support knowledge document 1004881.1 – Live Upgrade Software Patch Requirements (formerly 206844) also need to be installed. The latest packages and patches ensure that you have all the latest bug fixes and new features in the release. Ensure that you install all the patches that are relevant to your system before proceeding to create a new boot environment.

The [My Oracle Support](#) knowledge document 1004881.1 Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support describes how to remove old packages and install new packages, as well as lists the required patches. This section provides more description for the procedures described in knowledge document 1004881.1 – Live Upgrade Software Patch Requirements (formerly 206844).

You can install the packages by using the `liveupgrade20` command that is on the installation DVD or CD or use the `pkgadd` command. The `liveupgrade20` command requires Java software. If your system does not have Java software installed, then you need to use the `pkgadd` command to install the packages. See the My Oracle Support knowledge document for more information.

This section provides the following procedures:

- “Installing Patches Needed by Live Upgrade” on page 62
- “How to Install Live Upgrade With the `pkgadd` Command” on page 60
- “How to Install Live Upgrade With the Oracle Solaris Installation Program” on page 61

▼ How to Install Live Upgrade With the `pkgadd` Command

1 Remove existing Live Upgrade packages.

The three Live Upgrade packages, `SUNWluu`, `SUNWlur`, and `SUNWlucfg`, comprise the software needed to upgrade or patch by using Live Upgrade. These packages include existing software, new features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading or patching to the target release

fails. The `SUNWlucfg` package is new starting with the Solaris 10 8/07 release. If you are using Live Upgrade packages from a release previous to Solaris 10 8/07, you do not need to remove this package.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

2 Install the packages in the following order.

```
# pkgadd -d path-to-packages SUNWlucfg SUNWlur SUNWluu
```

path-to-packages Specifies the absolute path to the software packages.

3 (Optional) Verify that the package has been installed successfully.

```
# pkgchk -v SUNWlucfg SUNWlur SUNWluu
```

▼ How to Install Live Upgrade With the Oracle Solaris Installation Program

Using the `liveupgrade20` command that is on the installation DVD or CD requires Java software. If your system does not have Java software installed, then you need to use the `pkgadd` command to install the packages. See [“How to Install Live Upgrade With the `pkgadd` Command” on page 60](#) for more information.

Note – This procedure assumes that the system is running Solaris Volume Manager. For detailed information about managing removable media with the Solaris Volume Manager, refer to [System Administration Guide: Devices and File Systems](#).

1 Remove existing Live Upgrade packages.

The three Live Upgrade packages, `SUNWluu`, `SUNWlur`, and `SUNWlucfg`, comprise the software needed to upgrade or patch by using Live Upgrade. These packages include existing software, new features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading or patching to the target release fails. The `SUNWlucfg` package is new starting with the Solaris 10 8/07 release. If you are using Live Upgrade packages from a release previous to Solaris 10 8/07, you do not need to remove this package.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

2 Insert the Oracle Solaris Operating System DVD or Oracle Solaris Software - 2 CD.

3 Run the installer for the media you are using.

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

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

The Oracle Solaris installation program GUI is displayed. If you are using a script, you can prevent the GUI from displaying by using the `-noconsole` and `-nodisplay` options.

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

```
% ./installer
```

The Oracle Solaris installation program GUI is displayed.

- 4 **From the Select Type of Install panel, select Custom.**
- 5 **On the Locale Selection panel, select the language to be installed.**
- 6 **Choose the software to install.**
 - For the DVD, on the Component Selection panel, click Next to install the packages.
 - For the CD, on the Product Selection panel, select Default Install for Live Upgrade and click the other software choices to deselect them.
- 7 **Follow the directions on the Oracle Solaris installation program panels to install the software.**

You are ready to install the required patches.

Installing Patches Needed by Live Upgrade



Caution – Correct operation of Live Upgrade requires that a limited set of patch revisions be installed for a particular OS version. Before installing or running Live Upgrade, you are required to install these patches.

The patches listed in the knowledge document 1004881.1 – Live Upgrade Software Patch Requirements (formerly 206844) are subject to change at any time. These patches potentially fix defects in Live Upgrade, as well as fix defects in components that Live Upgrade depends on. If you experience any difficulties with Live Upgrade, please check and make sure that you have the latest Live Upgrade patches installed.

x86 only – If this set of patches is not installed, Live Upgrade fails and you might see the following error message. If you don't see the following error message, necessary patches still might not be installed. Always verify that all patches listed on the MyOracleSupport knowledge document have been installed before attempting to install Live Upgrade.

```
ERROR: Cannot find or is not
executable: </sbin/biosdev>.
ERROR: One or more patches required by
Live Upgrade has not been installed.
```

Ensure you have the most recently updated patch list by consulting <http://support.oracle.com>. Search for the knowledge document 1004881.1 – Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

If you are running the Solaris 8 or Solaris 9 OS, you might not be able to run the Live Upgrade installer. These releases do not contain the set of patches needed to run the Java 2 runtime environment. You must have the recommended patch cluster for the Java 2 runtime environment that is recommended to run the Live Upgrade installer and install the packages. To install the Live Upgrade packages, use the `pkgadd` command. Or install, for the Java 2 runtime environment, the recommended patch cluster. The patch cluster is available at <http://support.oracle.com> (My Oracle Support).

▼ How to Install Required Patches

- 1 If you are storing the patches on a local disk, create a directory such as `/var/tmp/lupatches`.
- 2 From the My Oracle Support web site, obtain the list of patches.
- 3 Change to the patch directory, for example, `lupatches`.

```
# cd /var/tmp/lupatches
```

- 4 Install the patches with the `patchadd` command.

```
# patchadd path-to-patches patch-ID patch-ID
```

patch-ID is the patch number or numbers. Separate multiple patch names with a space.

Note – The patches need to be applied in the order specified in the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

- 5 **Reboot the system if necessary.**

Certain patches require a reboot to be effective.

x86 only: Rebooting the system is required or Live Upgrade fails.

init 6

You now have the packages and patches necessary for a successful creation of a new boot environment.

Creating a New Boot Environment

Creating a boot environment provides a method of copying critical file systems from the active boot environment to a new boot environment. The `lucreate` command enables reorganizing a disk if necessary, customizing file systems, and copying the critical file systems to the new boot environment.

Before file systems are copied to the new boot environment, they can be customized so that critical file system directories are either merged into their parent directory or split from their parent directory. User-defined (shareable) file systems are shared between boot environments by default. But shareable file systems can be copied if needed. Swap, which is a shareable volume, can be split and merged also. For an overview of critical and shareable file systems, see [“File System Types” on page 22](#).

Note – This chapter describes Live Upgrade for UFS file systems. For procedures for migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see [Chapter 12](#), [“Creating a Boot Environment for ZFS Root Pools.”](#)

The `lucreate` command that is used with the `-m` option specifies which file systems and the number of file systems to be created in the new boot environment. You must specify the exact number of file systems you want to create by repeating this option. For example, a single use of the `-m` option specifies where to put all the file systems. You merge all the file systems from the original boot environment into the one file system that is specified by the `-m` option. If you specify the `-m` option twice, you create two file systems. When using the `-m` option to create file systems, follow these guidelines:

- You must specify one `-m` option for the root (`/`) file system for the new boot environment. If you run `lucreate` without the `-m` option, the Configuration menu is displayed. The Configuration menu enables you to customize the new boot environment by redirecting files onto new mount points.
- Any critical file systems that exist in the current boot environment and are not specified in a `-m` option are merged into the next highest level file system created.
- Only the file systems that are specified by the `-m` option are created on the new boot environment. If your current boot environment contains multiple file systems and you want to have the same number of file systems in the new boot environment created, you must

specify one `-m` option for each file system to be created. For example, if you have file systems for root (`/`), `/opt`, and `/var`, you would use one `-m` option for each file system on the new boot environment.

- Do not duplicate a mount point. For example, you cannot have two root (`/`) file systems.

▼ How to Create a Boot Environment for the First Time

- To create the new boot environment, type:

```
# lucreate [-A 'BE-description'] -c BE-name \
-m mountpoint:device[,metadevice]:fs-options [-m ...] -n BE-name
-A 'BE-description'
```

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (`BE-name`). The description can be any length and can contain any characters.

`-c BE-name`

Assigns the name `BE-name` to the active boot environment. This option is not required and is only used when the first boot environment is created. If you run `lucreate` for the first time and you omit the `-c` option, the software creates a default name for you.

The default name is chosen according to the following criteria:

- If the physical boot device can be determined, then the base name of the physical boot device is used to name the current boot environment.

For example, if the physical boot device is `/dev/dsk/c0t0d0s0`, then the current boot environment is given the name `c0t0d0s0`.

- If the physical boot device cannot be determined, then names from the `uname` command with the `-s` and `-r` options are combined to produce the name.

For example, if `uname -s` returns the OS name of SunOS and `uname -r` returns the release name of 5.9, then the name `SunOS5.9` is given to the current boot environment.

- If neither of these commands can determine the name, then the name `current` is used to name the current boot environment.

Note – If you use the `-c` option after the first boot environment creation, the option is ignored or an error message is displayed.

- If the name specified is the same as the current boot environment name, the option is ignored.
- If the name specified is different than the current boot environment name, then an error message is displayed and the creation fails. The following example shows a boot environment name that causes an error message.

```
# lucurr
c0t0d0s0
# lucreate -c /dev/dsk/clt1d1s1 -n newbe -m /:/dev/dsk/clt1d1s1:ufs
ERROR: current boot environment name is c0t0d0s0: cannot change
name using <-c clt1d1s1>
```

`-m mountpoint:device[,metadevice]:fs-options [-m ...]`

Specifies the file system configuration of the new boot environment in the `vfstab` file. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or `-` (hyphen), indicating a swap partition.
- *device* field can be one of the following:
 - The name of a disk device, of the form `/dev/dsk/cwt:xdysz`
 - The name of a Solaris Volume Manager volume, of the form `/dev/md/dsk/dnum`
 - The name of a Veritas Volume Manager volume, of the form `/dev/md/vxfs/dsk/dnum`
 - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs-options* field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap volume. The swap mount point must be a `-` (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)” on page 78](#).

`-n BE-name`

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

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–1 Creating a Boot Environment

In this example, the active boot environment is named `first_disk`. The mount points for the file systems are noted by using the `-m` option. Two file systems are created, `root (/)` and `/usr`. The new boot environment is named `second_disk`. A description, `mydescription`, is associated with the name `second_disk`. Swap, in the new boot environment `second_disk`, is automatically shared from the source, `first_disk`.

```
# lucreate -A 'mydescription' -c first_disk -m /:/dev/dsk/c0t4d0s0:ufs \
-m /usr:/dev/dsk/c0t4d0s3:ufs -n second_disk
```

▼ How to Create a Boot Environment and Merge File Systems

Note – You can use the `lucreate` command with the `-m` option to specify which file systems and the number of file systems to be created in the new boot environment. You must specify the exact number of file systems you want to create by repeating this option. For example, a single use of the `-m` option specifies where to put all the file systems. You merge all the file systems from the original boot environment into one file system. If you specify the `-m` option twice, you create two file systems.

● Type:

```
# lucreate -A 'BE_description' \
-m mountpoint:device[,metadevice]:fs_options \
-m [...] -m mountpoint:merged:fs_options -n BE_name
```

`-A BE_description`

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (`BE_name`). The description can be any length and can contain any characters.

`-m mountpoint:device[,metadevice]:fs_options [-m...]`

Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- `mountpoint` can be any valid mount point or `-` (hyphen), indicating a swap partition.
- `device` field can be one of the following:
 - The name of a disk device, of the form `/dev/dsk/cwt.xdysz`

- The name of a Solaris Volume Manager metadvice, of the form `/dev/md/dsk/dnum`
- The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/volume_name`
- The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- `fs_options` field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap volume. The swap mount point must be a `-` (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)” on page 78](#).

`-n BE_name`

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

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–2 Creating a Boot Environment and Merging File Systems

In this example, the file systems on the current boot environment are `root (/)`, `/usr`, and `/opt`. The `/opt` file system is combined with its parent file system `/usr`. The new boot environment is named `second_disk`. A description, `mydescription`, is associated with the name `second_disk`.

```
# lucreate -A 'mydescription' -c first_disk \  
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \  
-m /usr/opt:merged:ufs -n second_disk
```

▼ How to Create a Boot Environment and Split File Systems

Note – When creating file systems for a boot environment, the rules are identical to the rules for creating file systems for the Oracle Solaris OS. Live Upgrade cannot prevent you from making invalid configurations on critical file systems. For example, you could enter an `lucreate` command that would create separate file systems for `root (/)` and `/kernel`, which is an invalid division of the `root (/)` file system.

When splitting a directory into multiple mount points, hard links are not maintained across file systems. For example, if `/usr/stuff1/file` is hard linked to `/usr/stuff2/file`, and `/usr/stuff1` and `/usr/stuff2` are split into separate file systems, the link between the files no longer exists. `lucreate` issues a warning message and a symbolic link is created to replace the lost hard link.

● Type:

```
# lucreate [-A 'BE_description'] \
  -m mountpoint:device[,metadevice]:fs_options \
  -m mountpoint:device[,metadevice]:fs_options -n new_BE
-A 'BE_description'
```

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (*BE_name*). The description can be any length and contain any characters.

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or `-` (hyphen), indicating a swap partition.
- *device* field can be one of the following:
 - The name of a disk device, of the form `/dev/dsk/cwtxdysz`
 - The name of a Solaris Volume Manager metadevice, of the form `/dev/md/dsk/dnum`
 - The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/volume_name`
 - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap volume. The swap mount point must be a `-` (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)” on page 78](#).

```
-n BE_name
```

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

Example 4–3 Creating a Boot Environment and Splitting File Systems

In this example, the preceding command splits the root (/) file system over multiple disk slices in the new boot environment. Assume a source boot environment that has /usr, /var, and /opt on root (/): /dev/dsk/c0t0d0s0 /.

On the new boot environment, separate /usr, /var, and /opt, mounting these file systems on their own slices, as follows:

```
/dev/dsk/c0t1d0s0 /  
/dev/dsk/c0t1d0s1 /var  
/dev/dsk/c0t1d0s7 /usr  
/dev/dsk/c0t1d0s5 /opt
```

A description, mydescription, is associated with the boot environment name second_disk.

```
# lucreate -A 'mydescription' -c first_disk \  
-m /:/dev/dsk/c0t1d0s0:ufs -m /usr:/dev/dsk/c0t1d0s7:ufs \  
-m /var:/dev/dsk/c0t1d0s1:ufs -m /opt:/dev/dsk/c0t1d0s5:ufs \  
-n second_disk
```

Next Steps When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\).”](#)

▼ How to Create a Boot Environment and Reconfiguring Swap

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

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

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

- **Type:**

```
# lucreate [-A 'BE_description'] \  
-m mountpoint:device[,metadevice]:fs_options \  
-m -:device:swap -n BE_name
```

-A '*BE_description*'

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (*BE_name*). The description can be any length and can contain any characters.

-m *mountpoint:device[,metadevice]:fs_options* [-m...]

Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to -m can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
 - The name of a disk device, of the form */dev/dsk/cwtxdysz*
 - The name of a Solaris Volume Manager metadevice, of the form */dev/md/dsk/dnum*
 - The name of a Veritas Volume Manager volume, of the form */dev/vx/dsk/volume_name*
 - The keyword *merged*, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
 - *ufs*, which indicates a UFS file system.
 - *vxfs*, which indicates a Veritas file system.
 - *swap*, which indicates a swap volume. The swap mount point must be a – (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)”](#) on page 78.

-n *BE_name*

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

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

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–4 Creating a Boot Environment and Reconfiguring Swap

In this example, the current boot environment contains root (/) on */dev/dsk/c0t0d0s0* and swap is on */dev/dsk/c0t0d0s1*. The new boot environment copies root (/) to */dev/dsk/c0t4d0s0* and uses both */dev/dsk/c0t0d0s1* and */dev/dsk/c0t4d0s1* as swap slices. A description, *mydescription*, is associated with the boot environment name *second_disk*.

```
# lucreate -A 'mydescription' -c first_disk \  
-m /dev/dsk/c0t4d0s0:ufs -m -:/dev/dsk/c0t0d0s1:swap \  
-m -:/dev/dsk/c0t4d0s1:swap -n second_disk
```

These swap assignments are effective only after booting from `second_disk`. If you have a long list of swap slices, use the `-M` option. See [“How to Create a Boot Environment and Reconfigure Swap by Using a List” on page 72](#).

▼ How to Create a Boot Environment and Reconfigure Swap by Using a List

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

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

1 Create a list of swap slices to be used in the new boot environment.

The location and name of this file is user defined. In this example, the content of the `/etc/lu/swapslices` file is a list of devices and slices:

```
-:/dev/dsk/c0t3d0s2:swap  
-:/dev/dsk/c0t3d0s2:swap  
-:/dev/dsk/c0t4d0s2:swap  
-:/dev/dsk/c0t5d0s2:swap  
-:/dev/dsk/c1t3d0s2:swap  
-:/dev/dsk/c1t4d0s2:swap  
-:/dev/dsk/c1t5d0s2:swap
```

2 Type:

```
# lucreate [-A 'BE_description'] \  
-m mountpoint:device[,metadevice]:fs_options \  
-M slice_list -n BE_name
```

`-A 'BE_description'`

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (`BE_name`). The description can be any length and can contain any characters.

`-m mountpoint:device[,metadevice]:fs_options [-m...]`

Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or `-` (hyphen), indicating a swap partition.

- *device* field can be one of the following:
 - The name of a disk device, of the form `/dev/dsk/cwtxdysz`
 - The name of a Solaris Volume Manager metadvice, of the form `/dev/md/dsk/dnum`
 - The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/volume_name`
 - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap volume. The swap mount point must be a `-` (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)”](#) on page 78.

`-M slice_list`

List of `-m` options, which are collected in the file *slice_list*. Specify these arguments in the format that is specified for `-m`. Comment lines, which begin with a hash mark (`#`), are ignored. The `-M` option is useful when you have a long list of file systems for a boot environment. Note that you can combine `-m` and `-M` options. For example, you can store swap slices in *slice_list* and specify root (`/`) and `/usr` slices with `-m`.

The `-m` and `-M` options support the listing of multiple slices for a particular mount point. In processing these slices, `lucreate` skips any unavailable slices and selects the first available slice.

`-n BE_name`

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

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–5 Create a Boot Environment and Reconfiguring Swap By Using a List

In this example, swap in the new boot environment is the list of slices that are noted in the `/etc/lu/swapslices` file. A description, *mydescription*, is associated with the name `second_disk`.

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

▼ How to Create a Boot Environment and Copy a Shareable File System

If you want a shareable file system to be copied to the new boot environment, specify the mount point to be copied with the `-m` option. Otherwise, shareable file systems are shared by default, and maintain the same mount point in the `vfstab` file. Any updating that is applied to the shareable file system is available to both boot environments.

● Create the boot environment.

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
-m mountpoint:device[,metadevice]:fs_options -n BE_name
-A 'BE_description'
```

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (`BE_name`). The description can be any length and can contain any characters.

`-m mountpoint:device[,metadevice]:fs_options [-m...]`

Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- `mountpoint` can be any valid mount point or `-` (hyphen), indicating a swap partition.
- `device` field can be one of the following:
 - The name of a disk device, of the form `/dev/dsk/cwt:xdysz`
 - The name of a Solaris Volume Manager metadevice, of the form `/dev/md/dsk/dnum`
 - The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/volume_name`
 - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- `fs_options` field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap volume. The swap mount point must be a `-` (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)” on page 78](#).

-n *BE_name*

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

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\).”](#)

Example 4–6 Creating a Boot Environment and Copying a Shareable File System

In this example, the current boot environment contains two file systems, root (/) and /home. In the new boot environment, the root (/) file system is split into two file systems, root (/) and /usr. The /home file system is copied to the new boot environment. A description, *mydescription*, is associated with the boot environment name *second_disk*.

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \
-m /home:/dev/dsk/c0t4d0s4:ufs -n second_disk
```

▼ How to Create a Boot Environment From a Different Source

The `lucreate` command creates a boot environment that is based on the file systems in the active boot environment. If you want to create a boot environment based on a boot environment other than the active boot environment, use `lucreate` with the `-s` option.

Note – If you activate the new boot environment and need to fall back, you boot back to the boot environment that was last active, not the source boot environment.

● Create the boot environment.

```
# lucreate [-A 'BE_description'] -s source_BE_name
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

-A '*BE_description*'

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (*BE_name*). The description can be any length and can contain any characters.

-s *source_BE_name*

Specifies the source boot environment for the new boot environment. The source would not be the active boot environment.

-m *mountpoint:device[,metadevice]:fs_options* [-m...]

Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
 - The name of a disk device, of the form `/dev/dsk/cwt.xdysz`
 - The name of a Solaris Volume Manager metadvice, of the form `/dev/md/dsk/dnum`
 - The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/volume_name`
 - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap volume. The swap mount point must be a – (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)”](#) on page 78.

`-n BE_name`

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

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–7 Creating a Boot Environment From a Different Source

In this example, a boot environment is created that is based on the root (`/`) file system in the source boot environment named `third_disk`. `third_disk` is not the active boot environment. A description, `mydescription`, is associated with the new boot environment named `second_disk`.

```
# lucreate -A 'mydescription' -s third_disk \  
-m /:/dev/dsk/c0t4d0s0:ufs -n second_disk
```

▼ How to Create an Empty Boot Environment for a Flash Archive

The `lucreate` command creates a boot environment that is based on the file systems in the active boot environment. When using the `lucreate` command with the `-s` option, `lucreate` quickly creates an empty boot environment. The slices are reserved for the file systems that are specified, but no file systems are copied. The boot environment is named, but not actually

created until installed with a Flash Archive. When the empty boot environment is installed with an archive, file systems are installed on the reserved slices.

● Create the empty boot environment.

```
# lucreate -A 'BE_name' -s - \
-m mountpoint:device[,metadevice]:fs_options -n BE_name
-A 'BE_description'
```

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (BE_name). The description can be any length and can contain any characters.

-s -

Specifies that an empty boot environment be created.

-m *mountpoint:device[,metadevice]:fs_options* [-m...]

Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to -m can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
 - The name of a disk device, of the form */dev/dsk/cwt:xdysz*
 - The name of a Solaris Volume Manager metadevice, of the form */dev/md/dsk/dnum*
 - The name of a Veritas Volume Manager volume, of the form */dev/vx/dsk/volume_name*
 - The keyword *merged*, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
 - *ufs*, which indicates a UFS file system.
 - *vxfs*, which indicates a Veritas file system.
 - *swap*, which indicates a swap volume. The swap mount point must be a – (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)”](#) on page 78.

-n *BE_name*

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

Example 4–8 Creating an Empty Boot Environment for a Flash Archive

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

```
# lucreate -A 'mydescription' -s - \
-m /:/dev/dsk/c0t1d0s0:ufs -n second_disk
```

When creation of the empty boot environment is complete, a Flash archive can be installed and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

For an example of creating and populating an empty boot environment, see [“Example of Creating an Empty Boot Environment and Installing a Flash Archive”](#) on page 159.

▼ How to Create a Boot Environment With RAID-1 Volumes (Mirrors)

When you create a boot environment, Live Upgrade uses Solaris Volume Manager technology to create RAID-1 volumes. When creating a boot environment, you can use Live Upgrade to manage the following tasks.

- Remove a single-slice concatenation (submirror) from a RAID-1 volume (mirror). The contents can be saved to become the content of the new boot environment if necessary. Because the contents are not copied, the new boot environment can be quickly created. After the submirror is detached from a mirror, it is no longer part of the original mirror. Reads and writes to the submirror are no longer performed through the mirror.
- Create a boot environment that contains a mirror.
- Attach a single-slice concatenation to the newly created mirror.

Before You Begin To use the mirroring capabilities of Live Upgrade, you must create a state database and a state database replica. A state database stores information about disk about the state of your Solaris Volume Manager configuration.

- For information about creating a state database, see [Chapter 6, “State Database \(Overview\)”](#) in *Solaris Volume Manager Administration Guide*.
- For an overview of Solaris Volume Manager and the tasks that Live Upgrade can provide, see [“Creating a Boot Environment With RAID-1 Volume File Systems”](#) on page 27.
- For in-depth information about complex Solaris Volume Manager configurations that are not allowed when using Live Upgrade, see [Chapter 2, “Storage Management Concepts,”](#) in *Solaris Volume Manager Administration Guide*.

● **To create the new boot environment, type:**

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options [-m...] \
-n BE_name
```

```
-A 'BE_description'
```

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name *BE_name*. The description can be any length and can contain any characters.

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

Specifies the file systems' configuration of the new boot environment in the *vfstab*. The file systems that are specified as arguments to *-m* can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or – (hyphen), indicating a swap partition.
- *device* field can be one of the following:
 - The name of a disk device, of the form */dev/dsk/cwtxdysz*
 - The name of a Solaris Volume Manager volume, of the form */dev/md/dsk/dnum*
 - The name of a Veritas Volume Manager volume, of the form */dev/md/vxfs/dsk/dnum*
 - The keyword *merged*, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* can be one of the following types of file systems and keywords:
 - *ufs*, which indicates a UFS file system.
 - *vxfs*, which indicates a Veritas file system.
 - *swap*, which indicates a swap volume. The swap mount point must be a – (hyphen).
 - For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device.
 - *mirror* creates a RAID–1 volume or mirror on the specified device. In subsequent *-m* options, you must specify *attach* to attach at least one concatenation to the new mirror. The specified device must be correctly named. For example, a logical device name of */dev/md/dsk/d10* can serve as a mirror name. For more information about naming devices, see [“Overview of Solaris Volume Manager Components” in Solaris Volume Manager Administration Guide](#).
 - *detach* removes a concatenation from a volume that is associated with a specified mount point. The volume does not need to be specified.
 - *attach* attaches a concatenation to the mirror that is associated with a specified mount point. The physical disk slice that is specified is made into a single device concatenation for attaching to the mirror. To specify a concatenation to attach to

a disk, you append a comma and the name of that concatenation to the device name. If you omit the comma and the concatenation name, `lucreate` selects a free volume for the concatenation.

`lucreate` allows you to create only concatenations that contain a single physical slice. This command allows you to attach up to three concatenations to a mirror.

- `preserve` saves the existing file system and its content. This keyword enables you to bypass the copying process that copies the content of the source boot environment. Saving the content enables a quick creation of the new boot environment. For a particular mount point, you can use `preserve` with only one physical device. When you use `preserve`, `lucreate` checks that the device's content is suitable for a specified file system. This check is limited and cannot guarantee suitability.

The `preserve` keyword can be used with both a physical slice and a Solaris Volume Manager volume.

- If you use the `preserve` keyword when the UFS file system is on a physical slice, the content of the UFS file system is saved on the slice. In the following example of the `-m` option, the `preserve` keyword saves the content of the physical device `c0t0d0s0` as the file system for the mount point for the root (`/`) file system.

```
-m /:/dev/dsk/c0t0d0s0:preserve,ufs
```

- If you use the `preserve` keyword when the UFS file system is on a volume, the contents of the UFS file system are saved on the volume.

In the following example of the `-m` option, the `preserve` keyword saves the contents of the RAID-1 volume (mirror) `d10` as the file system for the mount point for the root (`/`) file system.

```
-m /:/dev/md/dsk/d10:preserve,ufs
```

In the following example of the `-m` option, a RAID-1 volume (mirror) `d10` is configured as the file system for the mount point for the root (`/`) file system. The single-slice concatenation `d20` is detached from its current mirror. `d20` is attached to mirror `d10`. The root (`/`) file system is preserved on submirror `d20`.

```
-m /:/dev/md/dsk/d10:mirror,ufs -m /:/dev/md/dsk/d20:detach,attach,preserve
```

```
-n BE_name
```

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

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–9 Creating a Boot Environment With a Mirror and Specifying Devices

In this example, the mount points for the file systems are specified by using the `-m` option.

- A description, `mydescription`, is associated with the name `another_disk`.
- `lucreate` configures a UFS file system for the mount point `root (/)`. A mirror, `d10`, is created. This mirror is the receptacle for the current boot environment's `root (/)` file system that is copied to the mirror `d10`. All data on the mirror `d10` is overwritten.
- Two slices, `c0t0d0s0` and `c0t1d0s0`, are submirrors, `d1` and `d2`. These two submirrors are added to mirror `d10`.
- The new boot environment is named `another_disk`.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,/dev/md/dsk/d1:attach \
-m /:/dev/dsk/c0t1d0s0,/dev/md/dsk/d2:attach -n another_disk
```

Example 4–10 Creating a Boot Environment With a Mirror and Not Specifying a Submirror Name

In this example, the mount points for the file systems are specified by using the `-m` option.

- A description, `mydescription`, is associated with the name `another_disk`.
- `lucreate` configures a UFS file system for the mount point `root (/)`. A mirror, `d10`, is created. This mirror is the receptacle for the current boot environment's `root (/)` file system that is copied to the mirror `d10`. All data on the mirror `d10` is overwritten.
- Two slices, `c0t0d0s0` and `c0t1d0s0`, are specified to be used as submirrors. The submirrors are not specified, but the `lucreate` command chooses names from a list of available volume names. These two submirrors are attached to mirror `d10`.
- The new boot environment is named `another_disk`.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0:attach \
-m /:/dev/dsk/c0t1d0s0:attach -n another_disk
```

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–11 Creating a Boot Environment and Detaching a Submirror

In this example, the mount points for the file systems are specified by using the `-m` option.

- A description, `mydescription`, is associated with the name `another_disk`.
- `lucreate` configures a UFS file system for the mount point `root (/)`. A mirror, `d10`, is created.

- Slice `c0t0d0s0` is removed from its current mirror. The slice is specified to be submirror `d1` and is added to mirror `d10`. The contents of the submirror, the root (`/`) file system, are saved and no copy occurs. Slice `c0t1d0s0` is submirror `d2` and is added to mirror `d10`.
- The new boot environment is named `another_disk`.

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0,/dev/md/dsk/d1:detach,attach,preserve \  
-m /:/dev/dsk/c0t1d0s0,/dev/md/dsk/d2:attach -n another_disk
```

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–12 Creating a Boot Environment, Detaching a Submirror, and Saving Its Contents

In this example, the mount points for the file systems are specified by using the `-m` option.

- A description, `mydescription`, is associated with the name `another_disk`.
- `lucreate` configures a UFS file system for the mount point root (`/`). A mirror, `d20`, is created.
- Slice `c0t0d0s0` is removed from its current mirror and added to the mirror `d20`. The name of the submirror is not specified. The contents of the submirror, the root (`/`) file system, are saved and no copy occurs.
- The new boot environment is named `another_disk`.

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d20:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0:detach,attach,preserve \  
-n another_disk
```

When the creation of the new boot environment is complete, the boot environment can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Example 4–13 Creating a Boot Environment With Two Mirrors

In this example, the mount points for the file systems are specified by using the `-m` option.

- A description, `mydescription`, is associated with the name `another_disk`.
- `lucreate` configures a UFS file system for the mount point root (`/`). A mirror, `d10`, is created. This mirror is the receptacle for the current boot environment's root (`/`) file system that is copied to the mirror `d10`. All data on the mirror `d10` is overwritten.
- Two slices, `c0t0d0s0` and `c0t1d0s0`, are submirrors `d1` and `d2`. These two submirrors are added to mirror `d10`.

- `lucreate` configures UFS file system for the mount point `/opt`. A mirror, `d11`, is created. This mirror is the receptacle for the current boot environment's `/opt` file system that is copied to the mirror `d11`. All data on the mirror `d11` is overwritten.
- Two slices, `c2t0d0s1` and `c3t1d0s1`, are submirrors `d3` and `d4`. These two submirrors are added to mirror `d11`.
- The new boot environment is named `another_disk`.

```
# lucreate -A 'mydescription' \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t0d0s0,/dev/md/dsk/d1:attach \
-m /:/dev/dsk/c0t1d0s0,/dev/md/dsk/d2:attach \
-m /opt:/dev/md/dsk/d11:ufs,mirror \
-m /opt:/dev/dsk/c2t0d0s1,/dev/md/dsk/d3:attach \
-m /opt:/dev/dsk/c3t1d0s1,/dev/md/dsk/d4:attach -n another_disk
```

When the creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

Creating a Boot Environment and Customizing the Content

The content of the file system on the new boot environment can be modified by using the inclusion and exclusion options. Directories and files are not copied to the new boot environment.

The inclusion and exclusion options appear in the `lucreate` command syntax as follows:

```
# lucreate -m mountpoint:device[,metadevice]:fs_options [-m ...] \
[-x exclude-dir] [-y include] \
[-Y include-list-file] \
[-f exclude-list-file] \
[-z filter-list] [-I] -n BE_name
```

`-x exclude-dir`

Excludes files and directories by not copying them to the new boot environment. You can use multiple instances of this option to exclude more than one file or directory.

exclude-dir is the name of the directory or file.

`-y include-dir`

Copies directories and files that are listed to the new boot environment. This option is used when you have excluded a directory but want to restore individual subdirectories or files.

include-dir is the name of the subdirectory or file to be included.

`-Y list-filename`

Copies directories and files from a list to the new boot environment. This option is used when you have excluded a directory but want to restore individual subdirectories or files.

- *list-filename* is the full path to a file that contains a list.
- The *list-filename* file must contain one file per line.
- If a line item is a directory, all subdirectories and files beneath that directory are included. If a line item is a file, only that file is included.

-f *list-filename*

Uses a list to exclude directories and files by not copying them to the new boot environment.

- *list-filename* is the full path to a file that contains a list.
- The *list-filename* file must contain one file per line.

-z *list-filename*

Uses a list to copy directories and files to the new boot environment. Each file or directory in the list is noted with a plus “+” or minus “-”. A plus indicates an included file or directory and the minus indicates an excluded file or directory.

- *list-filename* is the full path to a file that contains a list.
- The *list-filename* file must contain one file per line. A space must follow the plus or minus before the file name.
- If a line item is a directory and is indicated with a + (plus), all subdirectories and files beneath that directory are included. If a line item is a file and is indicated with a + (plus), only that file is included.

-I

Overrides the integrity check of system files. Use this option with caution.

To prevent you from removing important system files from a boot environment, `lucreate` runs an integrity check. This check examines all files that are registered in the system package database and stops the boot environment creation if any files are excluded. Use of this option overrides this integrity check. This option creates the boot environment more quickly, but might not detect problems.

Upgrading With Live Upgrade (Tasks)

This chapter explains how to use Live Upgrade to upgrade and activate an inactive boot environment.

Note – This chapter describes Live Upgrade for UFS file systems. The usage is the same for the `luupgrade` and `luactivate` commands for a ZFS boot environment. For procedures for migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see [Chapter 12, “Creating a Boot Environment for ZFS Root Pools.”](#)

This chapter contains the following sections:

- “Upgrading a Boot Environment (Task Map)” on page 85
- “Upgrading a Boot Environment” on page 86
- “Installing Flash Archives on a Boot Environment” on page 100
- “Activating a Boot Environment” on page 104

Upgrading a Boot Environment (Task Map)

TABLE 5–1 Task Map: Upgrading With Live Upgrade

Task	Description	For Instructions
Either upgrade a boot environment or install a flash archive.	<ul style="list-style-type: none"> ■ Upgrade the inactive boot environment with an OS image. ■ Install a flash archive on an inactive boot environment. 	<ul style="list-style-type: none"> ■ “Upgrading a Boot Environment” on page 86 ■ “Installing Flash Archives on a Boot Environment” on page 100
Activate an inactive boot environment.	Makes changes effective and switches the inactive boot environment to active.	“Activating a Boot Environment” on page 104

TABLE 5-1 Task Map: Upgrading With Live Upgrade (Continued)

Task	Description	For Instructions
(optional) Switch back if a failure occurs when activating.	Reactivates to the original boot environment if a failure occurs.	Chapter 6, “Failure Recovery: Falling Back to the Original Boot Environment (Tasks)”

Upgrading a Boot Environment

Use the `luupgrade` command to upgrade a boot environment. This section provides the procedure for upgrading an inactive boot environment from files that are located on the following media:

- NFS server
- Local file
- Local tape
- Local device, including DVD or CD

Guidelines for Upgrading

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

You can upgrade when RAID-1 volumes are installed, or if non-global zones are installed, or you can install a flash archive:

- You can upgrade an inactive boot environment that contains any combination of physical disk slices, Solaris Volume Manager volumes, or Veritas Volume Manager volumes. The slice that is chosen for the root (/) file system must be a single-slice concatenation that is included in a RAID-1 volume (mirror). For information about creating a boot environment with mirrored file systems, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)”](#) on page 78.

Note – If VxVM volumes are configured on your current system, the `lucreate` command can create a new boot environment. When the data is copied to the new boot environment, the Veritas file system configuration is lost and a UFS file system is created on the new boot environment.

- You can upgrade a system that has non-global zones installed with any installation program. For procedures about upgrading with non-global zones, see [Chapter 8, “Upgrading the Oracle Solaris OS on a System With Non-Global Zones Installed.”](#)

- Rather than upgrading, if you have created a flash archive, you could install the archive on an inactive boot environment. The new files overwrite critical file systems of the inactive boot environment, but shareable file systems are not changed. See [“Installing Flash Archives on a Boot Environment” on page 100](#).

Upgrading a System With Packages or Patches

You can use Live Upgrade to add patches and packages to a system. Live Upgrade creates a copy of the currently running system. This new boot environment can be upgraded or you can add packages or patches. When you use Live Upgrade, the only downtime the system incurs is that of a reboot. You can add patches and packages to a new boot environment with the `luupgrade` command.



Caution – When adding and removing packages or patches, Live Upgrade requires packages or patches that comply with the SVR4 advanced packaging guidelines. While Oracle packages conform to these guidelines, Oracle cannot guarantee the conformance of packages from third-party vendors. If a package violates these guidelines, the package can cause the package-addition software to fail or alter the active boot environment during an upgrade.

For more information about packaging requirements, see [Appendix C, “Additional SVR4 Packaging Requirements \(Reference\).”](#)

TABLE 5-2 Upgrading a Boot Environment With Packages and Patches

Type of Installation	Description	For More Information
Adding patches to a boot environment.	Create a new boot environment and use the <code>luupgrade</code> command with the <code>-t</code> option.	“Adding Patches to or Removing Patches From a Network Installation Image on a Boot Environment” on page 91
Adding packages to a boot environment.	Use the <code>luupgrade</code> command with the <code>-p</code> option.	“Adding Packages to or Removing Packages From a Network Installation Image on a Boot Environment” on page 90

▼ How to Upgrade a Network Installation Image on a Boot Environment

To upgrade by using this procedure, you must use a DVD or a network installation image. If the installation requires more than one CD, you must use the procedure [“How to Upgrade a Network Installation Image From Multiple CDs” on page 88](#).

Note – Starting with the Oracle Solaris 10 9/10 release, only a DVD is provided. Oracle Solaris Software CDs are no longer provided.

1 Install the Live Upgrade `SUNWlucfg`, `SUNWlur`, and `SUNWluu` packages on your system.

These packages must be from the release you are upgrading to. For a step-by-step procedure, see [“How to Install Live Upgrade With the `pkgadd` Command”](#) on page 60.

2 Indicate the boot environment to upgrade and the path to the installation software.

```
# luupgrade -u -n BE-name -s os-image-path
```

-u Upgrades a network installation image on a boot environment.

-n *BE-name* Specifies the name of the boot environment that is to be upgraded.

-s *os-image-path* Specifies the path name of a directory that contains a network installation image.

Example 5-1 Upgrading a Network Installation Image on a Boot Environment From DVD Media

In this example, the `second_disk` boot environment is upgraded by using DVD media. The `pkgadd` command adds the Live Upgrade packages from the release you are upgrading to.

```
# pkgadd -d /server/packages SUNWlucfg SUNWlur SUNWluu
# luupgrade -u -n second_disk -s /cdrom/cdrom0
```

Example 5-2 Upgrading a Network Installation Image on a Boot Environment From a Network Installation Image

In this example, the `second_disk` boot environment is upgraded. The `pkgadd` command adds the Live Upgrade packages from the release you are upgrading to.

```
# pkgadd -d /server/packages SUNWlucfg SUNWlur SUNWluu
# luupgrade -u -n second_disk \
-s /net/installmachine/export/Solaris_10/OS_image
```

▼ How to Upgrade a Network Installation Image From Multiple CDs

Note – Starting with the Oracle Solaris 10 9/10 release, only a DVD is provided. Oracle Solaris Software CDs are no longer provided.

Because the network installation image resides on more than one CD, you must use this upgrade procedure. Use the `luupgrade` command with the `-i` option to install any additional CDs.

1 Install the Live Upgrade `SUNWlucfg`, `SUNWlur`, and `SUNWluu` packages on your system.

These packages must be from the release you are upgrading to. For a step-by-step procedure, see [“How to Install Live Upgrade With the `pkgadd` Command” on page 60.](#)

2 Indicate the boot environment to upgrade and the path to the installation software.

```
# luupgrade -u -n BE-name -s os-image-path
```

- u Upgrades a network installation image on a boot environment.
- n *BE-name* Specifies the name of the boot environment that is to be upgraded.
- s *os-image-path* Specifies the path name of a directory that contains a network installation image.

3 When the installer is finished with the contents of the first CD, insert the second CD.

4 Run the installer on the second CD with menus or with text.

- To run the installer on the second CD with menus:

```
# luupgrade -i -n BE-name -s os-image-path
```

- To run the installer on the second CD with text and requires no user interaction.

```
# luupgrade -i -n BE-name -s os-image-path -O '-nodisplay -noconsole'
```

- i Installs additional CDs. The software looks for an installation program on the specified medium and runs that program. The installer program is specified with `-s`.
- n *BE-name* Specifies the name of the boot environment that is to be upgraded.
- s *os-image-path* Specifies the path name of a directory that contains an network installation image.
- O '-nodisplay -noconsole' (Optional) Runs the installer on the second CD in text mode and requires no user interaction.

5 Repeat [Step 3](#) and [Step 4](#) for each CD that you want to install.

Example 5-3 SPARC: Upgrading a Network Installation Image From Multiple CDs

In this example, the second `_disk` boot environment is upgraded and the installation image is on two CDs: the Oracle Solaris Software - 1 and the Oracle Solaris Software - 2 CDs. The `-u`

option determines if sufficient space for all the packages is on the CD set. The `-O` option with the `-nodisplay` and `-noconsole` options prevents the character user interface from displaying after the reading of the second CD. If you use these options, you are not prompted to type information.

Note that if you do not use the `-O` option with the `-nodisplay` and `-noconsole` options, the character user interface (CUI) is displayed. You do not need to use the CUI to perform Oracle Live Upgrade tasks.

1. Install the Live Upgrade packages from the release you are upgrading to.

```
# pkgadd -d /server/packages SUNWlucfg SUNWlur SUNWluu
```

2. Insert the Oracle Solaris Software - 1 CD and type the following command:

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

3. Insert the Oracle Solaris Software - 2 CD and type the following command:

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

Repeat this step for each CD that you need.

4. Repeat the previous step for each CD that you want to install.

Next Steps The boot environment is ready to be activated. See [“Activating a Boot Environment” on page 104](#).

Adding Packages to or Removing Packages From a Network Installation Image on a Boot Environment

This section describes how to remove packages from or add packages to a new boot environment.



Caution – When you are upgrading, adding and removing packages or patches, Live Upgrade requires packages or patches that comply with the SVR4 advanced packaging guidelines. While Oracle packages conform to these guidelines, Oracle cannot guarantee the conformance of packages from third-party vendors. If a package violates these guidelines, the package can cause the package-addition software to fail or can alter the active boot environment.

For more information about packaging requirements, see [Appendix C, “Additional SVR4 Packaging Requirements \(Reference\)”](#).

To remove a package or set of packages from a new boot environment, use the `-P` option.

```
# luupgrade -P -n BE-name package-name
```

-P	Indicates to remove the named package or packages from the boot environment.
-n <i>BE-name</i>	Specifies the name of the boot environment where the package is to be removed.
<i>package-name</i>	Specifies the names of the packages to be removed. Separate multiple package names with spaces.

To add a package or a set of packages to the new boot environment, use the **-P** option.

# <code>luupgrade -p -n BE-name -s /path-to-packages package-name</code>	
-p	Indicates to add packages to the boot environment.
-n <i>BE-name</i>	Specifies the name of the boot environment where the package is to be added.
-s <i>path-to-packages</i>	Specifies the path to a directory that contains the package or packages that are to be added.
<i>package-name</i>	Specifies the names of the package or packages to be added. Separate multiple package names with a space.

EXAMPLE 5-4 Adding Packages to a Network Installation Image on a Boot Environment

In this example, packages are removed and then added to the `second_disk` boot environment.

```
# luupgrade -P -n second_disk SUNWabc SUNWdef SUNWghi
# luupgrade -p -n second_disk -s /net/installmachine/export/packages \
SUNWijk SUNWlmn SUNWpkr
```

Adding Patches to or Removing Patches From a Network Installation Image on a Boot Environment

This section describes how to remove patches from or add patches to a new boot environment.



Caution – When you are adding and removing packages or patches, Live Upgrade requires packages or patches that comply with the SVR4 advanced packaging guidelines. While Oracle packages conform to these guidelines, Oracle cannot guarantee the conformance of packages from third-party vendors. If a package violates these guidelines, the package can cause the package-addition software to fail or can alter the active boot environment.

You cannot use Live Upgrade to patch a Oracle Solaris 10 inactive boot environment when the active boot environment is running the Solaris 8 or Solaris 9 OS. Live Upgrade will invoke the

patch utilities on the active boot partition to patch the inactive boot partition. The Solaris 8 and Solaris 9 patch utilities are unaware of Oracle Solaris Zones, Service Management Facility (SMF), and other enhancements in the Oracle Solaris 10 OS. Therefore the patch utilities fail to correctly patch an inactive Oracle Solaris 10 boot environment.

If you are using Live Upgrade to upgrade a system from the Solaris 8 or Solaris 9 OS to the Oracle Solaris 10 OS, you must first activate the Oracle Solaris 10 boot environment before patching. After the Oracle Solaris 10 boot environment is activated, you can either patch the active boot environment directly or set up another inactive boot environment and patch that one by using Live Upgrade. For an example of upgrading and patching from the Solaris 8 to the Oracle Solaris 10 release, see [Restrictions for Using Solaris Live Upgrade](#).

To remove a patch or set of patches from a new boot environment use the `-T` option.

```
# luupgrade -T -n BE-name patch_name
```

<code>-T</code>	Indicates to remove the named patch or patches from the boot environment.
<code>-n BE-name</code>	Specifies the name of the boot environment where the patch or patches are to be removed.
<code>patch-name</code>	Specifies the names of the patches to be removed. Separate multiple patch names with spaces.

To add a patch or a set of patches to the new boot environment, use the `-t` option.

```
# luupgrade -t -n BE-name -s /path-to-patches patch-name
```

<code>-t</code>	Indicates to add patches to the boot environment.
<code>-n BE-name</code>	Specifies the name of the boot environment where the patch is to be added.
<code>-s path-to-patches</code>	Specifies the path to the directory that contains the patches that are to be added.
<code>patch-name</code>	Specifies the names of the patch or patches that are to be added. Separate multiple patch names with a space.

EXAMPLE 5-5 Adding Patches to a Network Installation Image on a Boot Environment

In this example, patches are removed then added to the `second_disk` boot environment .

```
# luupgrade -T -n second_disk 222222-01
# luupgrade -t -n second_disk -s /net/installmachine/export/packages \
333333-01 444444-01
```

Checking Packages Installed on a Boot Environment

To check the integrity of the newly installed packages on the new boot environment, invoke the `pkgchk` command as follows:

```
# luupgrade -C -n BE-name -O "-v" package-name
```

`-C` Indicates to run the `pkgchk` command on the named packages.

`-n BE-name` Specifies the name of the boot environment where the check is to be performed.

`-O` Passes the options directly to the `pkgchk` command.

`package-name` Specifies the names of the packages to be checked. Separate multiple package names with spaces. If package names are omitted, the check is done on all packages in the specified boot environment.

`"-v"` Specifies to run the command in verbose mode.

EXAMPLE 5-6 Checking the Integrity of Packages on a Boot Environment

In this example, the packages `SUNWabc`, `SUNWdef`, and `SUNWghi` are checked to make sure they were installed properly and are not damaged.

```
# luupgrade -C -n second_disk SUNWabc SUNWdef SUNWghi
```

Upgrading by Using a JumpStart Profile

You can create a JumpStart profile to use with Live Upgrade. The following procedures enable you to create a profile, test the profile, and install by using the `luupgrade` command with the `-j` option.



Caution – When you install the Oracle Solaris OS with a flash archive, the archive and the installation media must contain identical OS versions. For example, if the archive is the Oracle Solaris 10 operating system and you are using DVD media, then you must use Oracle Solaris 10 DVD media to install the archive. If the OS versions do not match, the installation on the target system fails. Identical operating systems are necessary when you use the following keyword or command:

- `archive_location` keyword in a profile
- `luupgrade` command with the `-s`, `-a`, `-j`, and `-J` options

For more information see the following:

- [“How to Create a Profile to be Used by Live Upgrade” on page 94](#)
- [“How to Test a Profile to Be Used by Live Upgrade” on page 95](#)
- [“How to Upgrade With a Profile by Using Live Upgrade” on page 96](#)
- For creating a JumpStart profile, see [“Creating a Profile” in Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations](#)

▼ How to Create a Profile to be Used by Live Upgrade

This procedure shows you how to create a profile for use with Live Upgrade. You can use this profile to upgrade an inactive boot environment by using the `luupgrade` command with the `-j` option.

For procedures to use this profile, see the following sections:

- For an upgrade with a profile, see [“How to Upgrade With a Profile by Using Live Upgrade” on page 96](#).
- For a Flash Archive installation with a profile, see [“How to Install a Flash Archive With a Profile” on page 102](#).

1 Use a text editor to create a text file.

Name the file descriptively. Ensure that the name of the profile reflects how you intend to use the profile to install the Oracle Solaris software on a system. For example, you might name this profile `upgrade_Solaris_10`.

2 Add profile keywords and values to the profile.

Only the upgrade keywords in [Table 5–3](#) and [Table 5–4](#) can be used in a Live Upgrade profile.

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

3 Save the profile in a directory on the local system.

4 Ensure that `root` owns the profile and that the permissions are set to `644`.

Example 5–7 Creating a Live Upgrade Profile

In this example, a profile provides the upgrade parameters. This profile is to be used to upgrade an inactive boot environment with the Live Upgrade `luupgrade` command and the `-u` and `-j` options. This profile adds a package and a cluster. A regional locale and additional locales are also added to the profile. If you add locales to the profile, make sure that you have created a boot environment with additional disk space.

```
# profile keywords      profile values
# -----
```

install_type	upgrade
package	SUNWxwman add
cluster	SUNWCacc add
geo	C_Europe
locale	zh_TW
locale	zh_TW.BIG5
locale	zh_TW.UTF-8
locale	zh_HK.UTF-8
locale	zh_HK.BIG5HK
locale	zh
locale	zh_CN.GB18030
locale	zh_CN.GBK
locale	zh_CN.UTF-8

Example 5–8 Creating a Live Upgrade Profile to Install a Differential Archive

The following example of a profile is to be used by Live Upgrade to install a differential archive on a clone system. Only files that are specified by the differential archive are added, deleted, or changed. The flash archive is retrieved from an NFS server. Because the image was built by the original master system, the clone system is not checked for a valid system image. This profile is to be used with the Live Upgrade `luupgrade` command and the `-u` and `-j` options.

# profile keywords	profile values
# -----	-----
install_type	flash_update
archive_location	nfs installserver:/export/solaris/archive/solarisarchive
no_master_check	

For information about how to use the `luupgrade` command to install the differential archive, see [“How to Install a Flash Archive With a Profile” on page 102](#).

Next Steps If you want to test the profile, see [“How to Test a Profile to Be Used by Live Upgrade” on page 95](#).

▼ How to Test a Profile to Be Used by Live Upgrade

After you create a profile, use the `luupgrade` command to test the profile. By looking at the installation output that is generated by `luupgrade`, you can quickly determine if a profile works as you intended.

● Test the profile.

```
# luupgrade -u -n BE-name -D -s os-image-path -j profile-path
```

-u	Upgrades an operating system image on a boot environment.
-n <i>BE-name</i>	Specifies the name of the boot environment that is to be upgraded.
-D	Uses the selected boot environment's disk configuration to test the profile options that are passed with the <code>-j</code> option.

- s *os-image-path* Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.
- j *profile-path* Path to a profile that is configured for an upgrade. The profile must be in a directory on the local machine.

Example 5-9 Testing a Profile by Using Live Upgrade

In the following example, the profile is named `Flash_profile`. The profile is successfully tested on the inactive boot environment that is named `second_disk`.

```
# luupgrade -u -n ulb08 -D -s /net/installsvr/export/u1/combined.ulwos \
-j /var/tmp/flash_profile
Validating the contents of the media /net/installsvr/export/u1/combined.ulwos.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains Solaris version 10.
Locating upgrade profile template to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live Upgrade requests.
Creating upgrade profile for BE second_disk.
Determining packages to install or upgrade for BE second_disk.
Simulating the operating system upgrade of the BE second_disk.
The operating system upgrade simulation is complete.
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of the
upgrade operation.
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of
cleanup operations required.
The Solaris upgrade of the boot environment second_disk is complete.
```

You can now use the profile to upgrade an inactive boot environment.

▼ How to Upgrade With a Profile by Using Live Upgrade

This procedure provides step-by-step instructions for upgrading an OS by using a profile.

If you want to install a flash archive by using a profile, see [“How to Install a Flash Archive With a Profile” on page 102](#).

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



Caution – When you install the Oracle Solaris OS with a flash archive, the archive and the installation media must contain identical OS versions. For example, if the archive is the Oracle Solaris 10 operating system and you are using DVD media, then you must use Oracle Solaris 10 DVD media to install the archive. If the OS versions do not match, the installation on the target system fails. Identical operating systems are necessary when you use the following keyword or command:

- `archive_location` keyword in a profile
- `luupgrade` command with the `-s`, `-a`, `-j`, and `-J` options

1 Install the Live Upgrade `SUNWlucfg`, `SUNWlur`, and `SUNWluu` packages on your system.

These packages must be from the release you are upgrading to. For step-by-step procedures, see [“How to Install Live Upgrade With the `pkgadd` Command” on page 60](#).

2 Create a profile.

See [“How to Create a Profile to be Used by Live Upgrade” on page 94](#) for a list of upgrade keywords that can be used in a Live Upgrade profile.

3 Upgrade the OS by using a profile.

```
# luupgrade -u -n BE-name -s os-image-path -j profile-path
```

- | | |
|-------------------------------|---|
| <code>-u</code> | Upgrades an operating system image on a boot environment. |
| <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 directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory. |
| <code>-j profile-path</code> | Path to a profile. The profile must be in a directory on the local machine. For information about creating a profile, see “How to Create a Profile to be Used by Live Upgrade” on page 94 . |

Example 5–10 Upgrading a Boot Environment by Using a Custom JumpStart Profile

In this example, the `second_disk` boot environment is upgraded by using a profile. The `-j` option is used to access the profile. The boot environment is then ready to be activated. For information about how to create a profile, see [“How to Create a Profile to be Used by Live Upgrade” on page 94](#). The `pkgadd` command adds the Live Upgrade packages from the release you are upgrading to.

```
# pkgadd -d /server/packages SUNWlucfg SUNWlur SUNWluu
# luupgrade -u -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

Next Steps The boot environment is ready to be activated. See [“Activating a Boot Environment” on page 104.](#)

JumpStart Keyword Values

TABLE 5-3 Keywords for an Initial Archive Creation

Keywords	Description	Reference
(Required) <code>install_type</code>	Defines whether to upgrade the existing Oracle Solaris environment on a system or install a flash archive on the system. Use the following values with this keyword: <ul style="list-style-type: none">■ <code>upgrade</code> for an upgrade■ <code>flash_install</code> for a flash archive installation■ <code>flash_update</code> for a flash archive differential installation	For a description of all the values for this keyword, see “install_type Profile Keyword (UFS and ZFS)” in Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations.
(Required for a Flash Archive) <code>archive_location</code>	Retrieves a flash archive from a designated location.	For a list of values that can be used with this keyword, see “archive_location Keyword” in Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations.
(Optional) <code>cluster</code> (adding or deleting clusters)	Designates whether a cluster is to be added or deleted from the software group that is to be installed on the system.	For a list of values that can be used with this keyword, see “cluster Profile Keyword (Adding Software Groups) (UFS and ZFS)” in Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations.
(Optional) <code>geo</code>	Designates the regional locale or locales that you want to install on a system or to add when upgrading a system.	For a list of values that can be used with this keyword, see “geo Profile Keyword (UFS and ZFS)” in Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations.
(Optional) <code>local_customization</code>	Before you install a flash archive on a clone system, you can create custom scripts to preserve local configurations on the clone system. The <code>local_customization</code> keyword designates the directory where you have stored these scripts. The value is the path to the script on the clone system.	For information about predeployment and postdeployment scripts, see “Creating Customization Scripts” in Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation).

TABLE 5-3 Keywords for an Initial Archive Creation (Continued)

Keywords	Description	Reference
(Optional) <code>locale</code>	Designates the locale packages you want to install or add when upgrading.	For a list of values that can be used with this keyword, see “ locale Profile Keyword (UFS and ZFS) ” in <i>Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations</i> .
(Optional) <code>package</code>	Designates whether a package is to be added to or deleted from the software group that is to be installed on the system.	For a list of values that can be used with this keyword, see “ package Profile Keyword (UFS and ZFS) ” in <i>Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations</i> .

TABLE 5-4 Keywords for a Differential Archive Creation

Keywords	Description	Reference
(Required) <code>install_type</code>	Defines the installation to install a flash archive on the system. The value for a differential archive is <code>flash_update</code> .	For a description of all the values for this keyword, see “ install_type Profile Keyword (UFS and ZFS) ” in <i>Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations</i> .
(Required) <code>archive_location</code>	Retrieves a flash archive from a designated location.	For a list of values that can be used with this keyword, see “ archive_location Keyword ” in <i>Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations</i> .
(Optional) <code>forced_deployment</code>	Forces the installation of a flash archive differential archive onto a clone system that is different than the software expects. If you use <code>forced_deployment</code> , all new files are deleted to bring the clone system to the expected state. If you are not certain that you want files to be deleted, use the default, which protects new files by stopping the installation.	For more information about this keyword, see “ forced_deployment Profile Keyword (Installing Flash Archive Differential Archives) ” in <i>Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations</i> .
(Optional) <code>local_customization</code>	Before you install a flash archive on a clone system, you can create custom scripts to preserve local configurations on the clone system. The <code>local_customization</code> keyword designates the directory where you have stored these scripts. The value is the path to the script on the clone system.	For information about predeployment and postdeployment scripts, see “ Creating Customization Scripts ” in <i>Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation)</i> .

TABLE 5-4 Keywords for a Differential Archive Creation (Continued)

Keywords	Description	Reference
(Optional) no_content_check	When installing a clone system with a flash archive differential archive, you can use the no_content_check keyword to ignore file-by-file validation. File-by-file validation ensures that the clone system is a duplicate of the master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.	For more information about this keyword, see “no_content_check Profile Keyword (Installing Flash Archives)” in Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations .
(Optional) no_master_check	When installing a clone system with a flash archive differential archive, you can use the no_master_check keyword to ignore a check of files. Clone system files are not checked. A check would ensure the clone was built from the original master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.	For more information about this keyword, see “no_master_check Profile Keyword (Installing Flash Archives)” in Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations .

Installing Flash Archives on a Boot Environment

This section provides the procedure for using Live Upgrade to install Flash Archives. Installing a Flash Archive overwrites all files on the new boot environment except for shared files. Archives are stored on the following media:

- HTTP server
- FTP server – Use this path from the command line only
- NFS server
- Local file
- Local tape
- Local device, including DVD or CD

Note the following issues with installing and creating a Flash Archive.

When you install the Oracle Solaris OS with a flash archive, the archive and the installation media must contain identical OS versions. If the OS versions do not match, the installation on the target system fails. Identical operating systems are necessary when you use the following keyword or command:

- archive_location keyword in a profile
- luupgrade command with the -s, -a, -j, and -J options

A flash archive cannot be properly created when a non-global zone is installed. The Solaris Flash feature is not compatible with the Oracle Solaris Zones feature. If you create a Flash Archive in a non-global zone or create an archive in a global zone that has non-global zones installed, the resulting archive does not install properly when the archive is deployed. For

example, if the archive is the Oracle Solaris 10 operating system and you are using DVD media, then you must use Oracle Solaris 10 DVD media to install the archive.

For examples of the correct syntax for paths that are associated with archive storage, see “[archive_location Keyword](#)” in *Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations*.

To use the Flash Archive installation feature, you install a master system and create the flash archive. For more information about creating an archive, see [Chapter 3, “Creating Flash Archives \(Tasks\)”](#), in *Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation)*.

▼ How to Install a Flash Archive on a Boot Environment

1 Install the Live Upgrade `SUNWlucfg`, `SUNWlur`, and `SUNWluu` packages on your system.

These packages must be from the release you are upgrading to. For a step-by-step procedure, see “[How to Install Live Upgrade With the `pkgadd` Command](#)” on page 60.

2 Install from the flash archive.

```
# luupgrade -f -n BE-name -s os-image-path -a archive
```

-f	Indicates to install an operating system from a Flash Archive.
-n <i>BE-name</i>	Specifies the name of the boot environment that is to be installed with an archive.
-s <i>os-image-path</i>	Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory. This OS image provides a miniroot that boots a minimal, bootable root (/) file system to facilitate the installation of the flash archive. The miniroot is not the image that is installed. The -a option provides the operating system image.
-a <i>archive</i>	Path to the flash archive when the archive is available on the local file system. The operating system image versions that are specified with the -s option and the -a option must be identical.

Example 5–11 Installing Flash Archives on a Boot Environment

In this example, an archive is installed on the `second_disk` boot environment. The archive is located on the local system. The -a option provides the operating system image. The operating system versions for the -s and -a options are both Oracle Solaris 10 8/11 releases. All files are overwritten on `second_disk` except shareable files. The `pkgadd` command adds the Live Upgrade packages from the release you are upgrading to.

```
# pkgadd -d /server/packages SUNWlucfg SUNWlur SUNWluu
# luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_10/OS_image \
-a /net/server/archive/10
```

Next Steps The boot environment is ready to be activated. See [“Activating a Boot Environment” on page 104](#).

▼ How to Install a Flash Archive With a Profile

This procedure provides the steps to install a Flash Archive or differential archive by using a profile.

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

1 Install the Live Upgrade SUNWlucfg, SUNWlur, and SUNWluu packages on your system.

These packages must be from the release you are upgrading to. For a step-by-step procedure, see [“How to Install Live Upgrade With the pkgadd Command” on page 60](#).

2 Create a profile.

See [“How to Create a Profile to be Used by Live Upgrade” on page 94](#) for a list of keywords that can be used in a Live Upgrade profile.

3 Install the flash archive with the profile.

```
# luupgrade -f -n BE-name -s os-image-path -j profile-path
```

- | | |
|-------------------------|--|
| -f | Indicates to install an operating system from a Flash Archive. |
| -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. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory. This OS image provides a miniroot that boots a minimal, bootable root (/) file system to facilitate the installation of the flash archive. The miniroot is not the image that is installed. The -j option provides the path to the profile that contains the flash archive operating system image. |
| -j <i>profile-path</i> | Path to a JumpStart profile that is configured for a flash installation. The profile must be in a directory on the local machine. The -s option's operating system version and the flash archive operating system version must be identical. |

Example 5–12 Install a Flash Archive on a Boot Environment With a Profile

In this example, a profile provides the location of the archive to be installed.

```
# profile keywords      profile values
# -----
install_type           flash_install
archive_location       nfs installserver:/export/solaris/flasharchive/solarisarchive
```

After creating the profile, you can run the `luupgrade` command and install the archive. The `-s` option provides a miniroot that boots a minimal, bootable root (`/`) file system to facilitate the installation of the flash archive. The `-j` option provides the path to the profile that contains the path to the flash archive operating system image. The `pkgadd` command adds the Live Upgrade packages from the release you are upgrading to.

```
# pkgadd -d /server/packages SUNWlucfg SUNWlur SUNWluu
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

Next Steps The boot environment is then ready to be activated. See [“Activating a Boot Environment” on page 104](#).

▼ How to Install a Flash Archive With a Profile Keyword

This procedure enables you to install a flash archive and use the `archive_location` keyword at the command line rather than from a profile file. You can quickly retrieve an archive without the use of a profile file.

1 Install the Live Upgrade `SUNWlucfg`, `SUNWlur`, and `SUNWluu` packages on your system.

These packages must be from the release you are upgrading to. For a step-by-step procedure, see [“How to Install Live Upgrade With the `pkgadd` Command” on page 60](#).

2 Install the flash archive.

```
# luupgrade -f -n BE-name -s os-image-path -J 'archive_location path-to-profile'
```

- | | |
|--------------------------------------|---|
| <code>-f</code> | Specifies to upgrade an operating system from a Flash Archive. |
| <code>-n <i>BE-name</i></code> | Specifies the name of the boot environment that is to be upgraded. |
| <code>-s <i>os-image-path</i></code> | Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, |

or it can be an NFS or UFS directory. This OS image provides a miniroot that boots a minimal, bootable root (/) file system to facilitate the installation of the flash archive. The miniroot is not the image that is installed. The -j option provides the path to the profile that contains the flash archive operating system image.

-J 'archive_location path-to-profile'

Specifies the archive_location profile keyword and the path to the JumpStart profile. The -s option's operating system version and the flash archive operating system version must be identical. For the keyword values, see “[archive_location Keyword](#)” in *Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations*.

The boot environment is ready to be activated. See “[Activating a Boot Environment](#)” on [page 104](#).

Example 5–13 Installing a Flash Archive By Using a Profile Keyword

In this example, an archive is installed on the second_disk boot environment. The -s option provides a miniroot that boots a minimal, bootable root (/) file system to facilitate the installation of the flash archive. The -j option provides the path to the Flash Archive operating system image. The -J option and the archive_location keywords are used to retrieve the archive. All files are overwritten on second_disk except shareable files. The pkgadd command adds the Live Upgrade packages from the release you are upgrading to.

```
# pkgadd -d /server/packages SUNWlucfg SUNWlur SUNWluu
# 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 [Chapter 6, “Failure Recovery: Falling Back to the Original Boot Environment \(Tasks\)”](#).

- See “[How to Activate a Boot Environment](#)” on [page 106](#) to activate a boot environment with the luactivate command.

Note – The first time you activate a boot environment, the `luactivate` command must be used.

- See [“How to Activate a Boot Environment and Synchronize Files” on page 107](#) to activate a boot environment and force a synchronization of files.

Note – Files are synchronized with the first activation. If you switch boot environments after the first activation, files are not synchronized.

- See [“x86: How to Activate a Boot Environment With the GRUB Menu” on page 109](#) to activate a boot environment with the GRUB, a feature of Oracle Solaris, menu on x86.

Note – A GRUB menu can facilitate switching from one boot environment to another. A boot environment appears in the GRUB menu after the first activation.

Requirements and Limitations for Activating a Boot Environment

To successfully activate a boot environment, that boot environment must meet the following conditions:

- The boot environment must have a status of Complete. To find out how to check status, see [“Displaying the Status of All Boot Environments” on page 122](#)
- If the boot environment is not the current boot environment, you cannot have mounted the partitions of that boot environment by using the `luumount` or `mount` commands. See the [`luumount\(1M\)` or `mount\(1M\)` man page](#).
- The boot environment that you want to activate cannot be involved in a comparison operation. For procedures, see [“Comparing Boot Environments” on page 124](#). To reconfigure swap, see [“How to Create a Boot Environment and Reconfiguring Swap” on page 70](#).
- 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.

x86 only – If you have an x86 based system, you can also activate with the GRUB menu. Note the following exceptions:

- If a boot environment was created with the Solaris 8, Solaris 9, or Solaris 10 3/05 release, the boot environment must always be activated with the `luactivate` command. These older boot environments do not display on the GRUB menu.
- The first time you activate a boot environment, you must use the `luactivate` command. The next time you boot, that boot environment's name is displayed in the GRUB main menu. You can thereafter switch to this boot environment by selecting the appropriate entry in the GRUB menu.

See [“x86: Activating a Boot Environment With the GRUB Menu”](#) on page 108.

▼ How to Activate a Boot Environment

The following procedure switches a new boot environment to become the currently running boot environment.

x86 only – If you have an x86 based system, you can also activate with the GRUB menu. Note the following exceptions:

- If a boot environment was created with the Solaris 8, Solaris 9, or Solaris 10 3/05 release, the boot environment must always be activated with the `luactivate` command. These older boot environments do not display on the GRUB menu.
- The first time you activate a boot environment, you must use the `luactivate` command. The next time you boot, that boot environment's name is displayed in the GRUB main menu. You can thereafter switch to this boot environment by selecting the appropriate entry in the GRUB menu.

See [“x86: Activating a Boot Environment With the GRUB Menu”](#) on page 108.

1 To activate the boot environment, type:

```
# /sbin/luactivate BE-name
```

BE-name Specifies the name of the boot environment that is to be activated

2 Reboot.

```
# init 6
```



Caution – Use only the `init` or `shutdown` commands to reboot. If you use the `reboot`, `halt`, or `uadmin` commands, the system does not switch boot environments. The last-active boot environment is booted again.

Example 5–14 Activating a Boot Environment

In this example, the `second_disk` boot environment is activated at the next reboot.

```
# /sbin/luactivate second_disk
# init 6
```

▼ How to Activate a Boot Environment and Synchronize Files

The first time you boot from a newly created boot environment, Live Upgrade software synchronizes the new boot environment with the boot environment that was last active. “Synchronize” means that certain critical system files and directories are copied from the last-active boot environment to the boot environment being booted. Live Upgrade does not perform this synchronization after the initial boot, unless you force synchronization with the `luactivate` command and the `-s` option.

x86 only – When you switch between boot environments with the GRUB menu, files also are not synchronized. You must use the following procedure to synchronize files.

For more information about synchronization, see [“Synchronizing Files Between Boot Environments” on page 54](#).

1 To activate the boot environment, type:

```
# /sbin/luactivate -s BE-name
```

-s Forces a synchronization of files between the last-active boot environment and the new boot environment. The first time that a boot environment is activated, the files between the boot environment are synchronized. With subsequent activations, the files are not synchronized unless you use the `-s` option.



Caution – Use this option with great care, because you might not be aware of or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Oracle Solaris 10 8/11 software on your current boot environment and booted back to a Solaris 9 release with a forced synchronization, files could be changed on the Solaris 9 release. Because files are dependent on the release of the OS, the boot to the Solaris 9 release could fail because the Oracle Solaris 10 8/11 files might not be compatible with the Solaris 9 files.

BE-name Specifies the name of the boot environment that is to be activated.

2 Reboot.

```
# init 6
```

Example 5–15 Activating a Boot Environment

In this example, the `second_disk` boot environment is activated at the next reboot and the files are synchronized.

```
# /sbin/luactivate -s second_disk
# init 6
```

x86: Activating a Boot Environment With the GRUB Menu

A GRUB menu provides an optional method of switching between boot environments. The GRUB menu is an alternative to activating (booting) with the `luactivate` command. The cautions and limitations are as follows:



Caution – After you have activated a boot environment, do not change the disk order in the BIOS. Changing the order might cause the GRUB menu to become invalid. If this problem occurs, changing the disk order back to the original state fixes the GRUB menu.

- The first time you activate a boot environment, you must use the `luactivate` command. The next time you boot, that boot environment's name is displayed in the GRUB main menu. You can thereafter switch to this boot environment by selecting the appropriate entry in the GRUB menu. For information about how to activate a boot environment, see [“How to Activate a Boot Environment”](#) on page 106.

- The first time you activate a boot environment, files are synchronized between the current boot environment and the new boot environment. With subsequent activations, files are not synchronized. When you switch between boot environments with the GRUB menu, files also are not synchronized. You can force a synchronization when using the `luactivate` command with the `-s` option. For information about how to activate a boot environment and synchronize files, see [“How to Activate a Boot Environment and Synchronize Files” on page 107](#).
- If a boot environment was created with the Solaris 8, Solaris 9, or Solaris 10 3/05 release, the boot environment must always be activated with the `luactivate` command. These older boot environments do not display on the GRUB menu. For information about how to activate a boot environment, see [“How to Activate a Boot Environment” on page 106](#).
- The `menu.lst` file contains the information that is displayed in the GRUB menu. You can revise this file for the following reasons:
 - To add to the GRUB menu entries for operating systems other than the Oracle Solaris OS. For more information, see [“GRUB Based Booting \(Overview\)” in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*](#)
 - To customize booting behavior. For example, you could change booting to verbose mode or change the default time that automatically boots the OS. For more information, see [“Booting an x86 Based System by Using GRUB \(Task Map\)” in *Oracle Solaris Administration: Basic Administration*](#).

Note – If you want to change the GRUB menu, you need to locate the `menu.lst` file. For step-by-step instructions, see [Chapter 13, “Managing the Oracle Solaris Boot Archives \(Tasks\)” in *Oracle Solaris Administration: Basic Administration*](#).



Caution – Do not use the GRUB `menu.lst` file to modify Live Upgrade entries. Modifications could cause Live Upgrade to fail. Although you can use the `menu.lst` file to customize booting behavior, the preferred method for customization is to use the `eeprom` command. If you use the `menu.lst` file to customize, the Oracle Solaris OS entries might be modified during a software upgrade. Changes to the file could be lost.

▼ **x86: How to Activate a Boot Environment With the GRUB Menu**

You can switch between two boot environments with the GRUB menu. Note the following limitations:

- The first activation of a boot environment must be done with the `luactivate` command. After the initial activation, the boot environment is displayed on the GRUB menu. The boot environment can then be booted from the GRUB menu.
- Switching to a boot environment with the GRUB menu bypasses synchronization. For more information about synchronizing files, see link [“Forcing a Synchronization Between Boot Environments”](#) on page 55.
- If a boot environment was created with the Solaris 8, Solaris 9, or Solaris 10 3/05 release, the boot environment must always be activated with the `luactivate` command. These older boot environments are not displayed on the GRUB menu.

1 Reboot the system.

`init 6`

The GRUB main menu is displayed.

In this example, the two operating systems are listed, Oracle Solaris and `second_disk`, which is a Live Upgrade boot environment. The `failsafe` entries are for recovery, if for some reason the primary OS does not boot.

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris                               |
|Solaris failsafe                      |
|second_disk                          |
|second_disk failsafe                 |
+-----+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

2 Use the arrow key to select the desired boot environment and press Enter.

The selected boot environment is booted and becomes the active boot environment.

Failure Recovery: Falling Back to the Original Boot Environment (Tasks)

This chapter explains how to recover from an activation failure.

Note – This chapter describes Live Upgrade for UFS file systems. The usage for the `luactivate` command for a ZFS boot environment is the same. For procedures for migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see [Chapter 12, “Creating a Boot Environment for ZFS Root Pools.”](#)

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:**
 - “SPARC: To Fall Back Despite Successful New Boot Environment Activation” on page 112
 - “SPARC: To Fall Back From a Failed Boot Environment Activation” on page 112
 - “SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 113
- **For x86 based systems:**
 - “x86: To Fall Back Despite Successful New Boot Environment Activation With the GRUB Menu” on page 114
 - “x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu” on page 115
 - “x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu and the DVD or CD” on page 117

SPARC: Falling Back to the Original Boot Environment

You can fallback to the original boot environment by using one of these methods:

- [“SPARC: To Fall Back Despite Successful New Boot Environment Activation” on page 112](#)
- [“SPARC: To Fall Back From a Failed Boot Environment Activation” on page 112](#)
- [“SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 113](#)

▼ SPARC: To Fall Back Despite Successful New Boot Environment Activation

Use this procedure when you have successfully activated your new boot environment but are unhappy with the results.

- 1 **Activate the desired boot environment.**

```
# /sbin/luactivate BE-name
```

- 2 **Reboot.**

```
# init 6
```

The previous working boot environment becomes the active boot environment.

▼ SPARC: To Fall Back From a Failed Boot Environment Activation

- If you experience a failure while booting the new boot environment and can boot the original boot environment in single-user mode, use this procedure to fall back to the original boot environment.
- If you need to boot from media or a net installation image, see [“SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 113](#).

- 1 **At the OK prompt, boot the machine to single-user state from the Oracle Solaris Operating System DVD, Oracle Solaris Software - 1 CD, the network, or a local disk.**

```
OK boot device-name -s
```

device-name Specifies the name of devices from which the system can boot, use the format /dev/dsk/cwtxdysz, for example /dev/dsk/c0t0d0s0.

- 2 **Activate the desired boot environment.**

```
# /sbin/luactivate BE-name
```


- If this command fails to display a prompt, proceed to “[SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image](#)” on page 113.
- If the prompt is displayed, continue.

3 At the prompt, verify that you want to activate the original boot environment.

Do you want to fallback to activate boot environment <disk name>
(yes or no)? **yes**

A message displays that the fallback activation is successful.

4 Reboot.

init 6

The previous working boot environment becomes the active boot environment.

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

Use this procedure to boot from a DVD, CD, a net installation image or another disk that can be booted. 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 Oracle Solaris Operating System DVD, Oracle Solaris Software - 1 CD, the network, or a local disk:

OK **boot cdrom -s**

or

OK **boot net -s**

or

OK **boot device-name -s**

device-name Specifies the name of the disk and the slice where a copy of the operating system resides, use the format `/dev/dsk/cwtxdysz`, for example `/dev/dsk/c0t0d0s0`.

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

fsck device-name

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

mount device-name /mnt

- 4 **From the active boot environment root (/) slice, activate the previous working boot environment.**

```
# /mnt/sbin/luactivate
```

- 5 **Unmount the directory.**

```
# umount /mnt
```

- 6 **Reboot.**

```
# init 6
```

The previous working boot environment becomes the active boot environment.

x86: Falling Back to the Original Boot Environment

To fall back to the original boot environment, choose the procedure the best fits your circumstances.

- [“x86: To Fall Back Despite Successful New Boot Environment Activation With the GRUB Menu” on page 114](#)
- [“x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu” on page 115](#)
- [“x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu and the DVD or CD” on page 117](#)

▼ x86: To Fall Back Despite Successful New Boot Environment Activation With the GRUB Menu

Use this procedure when you have successfully activated your new boot environment but are dissatisfied with the results. You can quickly switch back to the original boot environment by using the GRUB.

Note – The boot environments that are being switched must be GRUB boot environments that were created with GRUB software. If a boot environment was created with the Solaris 8, Solaris 9, or Solaris 10 3/05 release, the boot environment is not a GRUB boot environment.

- 1 **Reboot the system.**

```
# init 6
```

The GRUB menu is displayed. The Oracle Solaris OS is the original boot environment. The `second_disk` boot environment was successfully activated and appears on the GRUB menu. The failsafe entries are for recovery if for some reason the primary entry does not boot.

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris                                     |
|Solaris failsafe                           |
|second_disk                                |
|second_disk failsafe                       |
+-----+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

- 2 To boot to the original boot environment, use the arrow key to select the original boot environment and press Return.

▼ x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu

If you experience a failure while booting, use the following procedure to fall back to the original boot environment. In this example, the GRUB menu is displayed correctly but the new boot environment is not bootable. The device is `/dev/dsk/c0t4d0s0`. The original boot environment, `c0t4d0s0`, becomes the active boot environment.



Caution – For the Solaris 10 3/05 release, the recommended action to fall back if the previous boot environment and new boot environment were on different disks included changing the hard disk boot order in the BIOS. **Starting with the Solaris 10 1/06 release**, changing the BIOS disk order is unnecessary and is strongly discouraged. Changing the BIOS disk order might invalidate the GRUB menu and cause the boot environment to become unbootable. If the BIOS disk order is changed, reverting the order back to the original settings restores system functionality.

Before You Begin

The boot environment must have been created with GRUB software. A boot environment that was created before the **Solaris 10 1/06 release** is not a GRUB boot environment. If you do not have a bootable GRUB boot environment, then go to the procedure, [“x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu and the DVD or CD”](#) on page 117.

- 1 To display the GRUB menu, reboot the system.

```
# init 6
```

The GRUB menu is displayed.

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris
|Solaris failsafe
|second_disk
|second_disk failsafe
+-----+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

2 From the GRUB menu, select the original boot environment.

3 Boot to single user mode by editing the GRUB menu.

a. Type e to display the GRUB edit menu.

```
root (hd0,2,a)
kernel /platform/i86pc/multiboot
module /platform/i86pc/boot_archive
```

b. Select the original boot environment's kernel entry by using the arrow keys.

c. Type e.

The kernel entry is displayed in the GRUB edit menu.

```
grub edit>kernel /boot/multiboot
```

d. Type -s and press Enter.

The following example notes the placement of the -s option.

```
grub edit>kernel /boot/multiboot -s
```

e. To begin the booting process in single user mode, type b.

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

```
# fsck mount-point
```

mount-point A root (/) file system that is known and reliable

5 Mount the original boot environment root slice to some directory (such as /mnt).

```
# mount device-name /mnt
```

device-name Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is provided in the form /dev/dsk/cwtxdysz.

- 6 From the active boot environment root slice, activate the previous working boot environment.

```
# /mnt/sbin/luactivate
```

- 7 Unmount the directory.

```
# umount /mnt
```

- 8 Reboot.

```
# init 6
```

The previous working boot environment becomes the active boot environment.

▼ x86: To Fall Back From a Failed Boot Environment Activation With the GRUB Menu and the DVD or CD

If you experience a failure while booting, use the following procedure to fall back to the original boot environment. In this example, the new boot environment was not bootable. Also, the GRUB menu does not display. The device is `/dev/dsk/c0t4d0s0`. The original boot environment, `c0t4d0s0`, becomes the active boot environment.



Caution – For the Solaris 10 3/05 release, the recommended action to fall back if the previous boot environment and new boot environment were on different disks included changing the hard disk boot order in the BIOS. **Starting with the Solaris 10 1/06 release**, changing the BIOS disk order is unnecessary and is strongly discouraged. Changing the BIOS disk order might invalidate the GRUB menu and cause the boot environment to become unbootable. If the BIOS disk order is changed, reverting the order back to the original settings restores system functionality.

- 1 Become superuser or assume equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see *Configuring RBAC (Task Map) in System Administration Guide: Security Services*.

- 2 Insert the Oracle Solaris Operating System for x86 Platforms DVD or the Oracle Solaris Software for x86 Platforms - 1 CD.

- 3 Boot from the DVD or CD.

```
# init 6
```

The GRUB menu is displayed.

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
```

```
+-----+
|Oracle Solaris 10 8/11                               |
|Oracle Solaris 10 8/11 Serial Console ttya           |
```

```
|Oracle Solaris 10 8/11 Serial Console ttyb (for lx50, v60x and v65x      |
+-----+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

4 Wait for the default option to boot or choose any option displayed.

The installation screen is displayed.

```
+-----+
|
|Select the type of installation you want to perform:
|
|      1 Solaris Interactive
|      2 Custom JumpStart
|      3 Solaris Interactive Text (Desktop session)
|      4 Solaris Interactive Text (Console session)
|      5 Apply driver updates
|      6 Single user shell
|
|      Enter the number of your choice followed by the <ENTER> key.
|      Alternatively, enter custom boot arguments directly.
|
|      If you wait 30 seconds without typing anything,
|      an interactive installation will be started.
|
+-----+
```

5 Choose the Single User Shell option.

The following message is displayed.

```
Do you wish to automatically update the boot archive? y /n
```

6 Type n.

```
Starting shell...
#
```

You are now in single user mode.

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

```
# fsck mount-point
```

mount-point A root (/) file system that is known and reliable

8 Mount the original boot environment root slice to some directory (such as /mnt).

```
# mount device-name /mnt
```

device-name Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is provided in the form of /dev/dsk/cwtxdysz.

9 From the active boot environment root slice, activate the previous working boot environment.

```
# /mnt/sbin/luactivate
```

```
Do you want to fallback to activate boot environment c0t4d0s0
```

```
(yes or no)? yes
```

10 Unmount the directory.

```
# umount device_name
```

11 Reboot.

```
# init 6
```

The previous working boot environment becomes the active boot environment.

Maintaining Live Upgrade Boot Environments (Tasks)

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

Note – This chapter describes Live Upgrade, a feature of Oracle Solaris, for UFS file systems. The usage for the maintenance for a ZFS boot environment is the same. For procedures for migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see [Chapter 12, “Creating a Boot Environment for ZFS Root Pools.”](#)

This chapter contains the following sections:

- [“Overview of Live Upgrade Maintenance” on page 122](#)
- [“Displaying the Status of All Boot Environments” on page 122](#)
- [“Updating a Previously Configured Boot Environment” on page 123](#)
- [“Canceling a Scheduled Create, Upgrade, or Copy Job” on page 124](#)
- [“Comparing Boot Environments” on page 124](#)
- [“Deleting an Inactive Boot Environment” on page 125](#)
- [“Displaying the Name of the Active Boot Environment” on page 126](#)
- [“Changing the Name of a Boot Environment” on page 126](#)
- [“Adding or Changing a Description Associated With a Boot Environment Name” on page 127](#)
- [“Viewing the Configuration of a Boot Environment” on page 128](#)

Overview of Live Upgrade Maintenance

TABLE 7-1 Overview of Possible Live Upgrade Maintenance Tasks

Task	Description	For Instructions
View status.	<ul style="list-style-type: none">■ View whether a boot environment is active, being activated, scheduled to be activated, or in the midst of a comparison.■ Compare the active and inactive boot environments.■ Display the name of the active boot environment.■ View the configurations of a boot environment.	<ul style="list-style-type: none">■ “Displaying the Status of All Boot Environments” on page 122■ “Comparing Boot Environments” on page 124■ “Displaying the Name of the Active Boot Environment” on page 126■ “Viewing the Configuration of a Boot Environment” on page 128
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 123
Modify a boot environment.	<ul style="list-style-type: none">■ Delete a boot environment.■ Change the name of a boot environment.■ Add or change a description that is associated with a boot environment name.	<ul style="list-style-type: none">■ “Deleting an Inactive Boot Environment” on page 125■ “Changing the Name of a Boot Environment” on page 126■ “Adding or Changing a Description Associated With a Boot Environment Name” on page 127
Cancel a scheduled job	<ul style="list-style-type: none">■ Cancel a scheduled create, update, or copy job.	■ “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 124

Maintenance Activities for Boot Environments

Displaying the Status of All Boot Environments

Use the `lustatus` command to display the information about the boot environment. If no boot environment is specified, the status information for all boot environments on the system is displayed.

The following details for each boot environment are displayed:

- Name – Name of each boot environment.
- Complete – Indicates that no copy or create operations are in progress. Also, the boot environment can be booted. Any current activity or failure in a create or upgrade operation causes a boot environment to be incomplete. For example, if a copy operation is in process or scheduled for a boot environment, that boot environment is considered incomplete.
- Active – Indicates whether this is the active boot environment.
- ActiveOnReboot – Indicates whether the boot environment becomes active on next reboot of the system.
- CopyStatus – Indicates whether the creation or copy of the boot environment is scheduled, active, or in the process of being upgraded. A status of SCHEDULED prevents you from performing Live Upgrade copy, rename, or upgrade operations.

In this example, the status for all boot environments is displayed. To display the status of a specific boot environment, issue the command with the boot environment's name.

```
# su
# lustatus
boot environment      Is      Active  Active  Can      Copy
Name                 Complete Now    OnReboot Delete   Status
-----
disk_a_S9             yes     yes     yes     no       -
disk_b_S10database    yes     no      no      yes      COPYING
disk_b_S9a            no      no      no      yes      -
```

Based on the sample output, you could not perform copy, rename, or upgrade operations on disk_b_S9a because it is not complete, nor on disk_b_S10database because a Live Upgrade operation is in progress.

Updating a Previously Configured Boot Environment

You can update the contents of a previously configured boot environment with the Copy menu or the lumake command. File systems from the active (source) boot environment are copied to the target boot environment. The data on the target is also destroyed. A boot environment must have the status Complete before you can copy from it. See [“Displaying the Status of All Boot Environments” on page 122](#) for more information about how 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 124](#).

The syntax for the lumake command is as follows:

```
# lumake -n BE-name [-s source-BE] [-t time] [-m email-address]
-n BE-name           Specifies the name of the boot environment that has file systems that are to
                     be replaced.
```

- s *source-BE* (Optional) Specifies the name of the source boot environment that contains the file systems to be copied to the target boot environment. If you omit this option, `lumake` uses the current boot environment as the source.
- t *time* (Optional) Setup a batch job to copy over file systems on a specified boot environment at a specified time. For information about how to format the time argument, see `at(1)` man page.
- m *email-address* (Optional) Enables you to send an email of the `lumake` output to a specified address on command completion. *email-address* is not checked. You can use this option only in conjunction with -t.

EXAMPLE 7-1 Updating a Previously Configured Boot Environment

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

```
# su
# lumake -n second_disk -s first_disk -m joe@anywhere.com
```

The files on `first_disk` are copied to `second_disk` and email is sent for notification. To find how to cancel a scheduled copy, see [“Canceling a Scheduled Create, Upgrade, or Copy Job” on page 124](#).

Canceling a Scheduled Create, Upgrade, or Copy Job

A boot environment's scheduled creation, upgrade, or copy job can be canceled just prior to the time the job starts. The job can be scheduled by the `lumake` command. At any time, only one job can be scheduled on a system.

To cancel a scheduled job, become superuser or assume an equivalent role and issue the `lumancel` command.

Comparing Boot Environments

Use the `lucompare` command 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 122](#).

The `lucompare` command generates a comparison of boot environments that includes the contents of any non-global zones.

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

The syntax for the `lucompare` command is as follows:

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

-i *infile* Compare files that are listed in *infile*. The files to be compared should have absolute file names. If the entry in the file is a directory, then comparison is recursive to the directory. Use either this option or **-t**, not both.

-t Compare only nonbinary files. This comparison uses the `file(1)` command on each file to determine whether the file is a text file. Use either this option or **-i**, not both.

-o *outfile* Redirect the output of differences to *outfile*.

BE-name Specifies the name of the boot environment that is compared to the active boot environment.

EXAMPLE 7-2 Comparing Boot Environments

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 the `ludelete` command to remove a boot environment. Note the following limitations.

- You cannot delete the active boot environment or the boot environment that is activated on the next reboot.
- The boot environment to be deleted must be complete. A complete boot environment is not participating in an operation that will change its status. “[Displaying the Status of All Boot Environments](#)” on page 122 describes how to determine a boot environment's status.
- You cannot delete a boot environment that has file systems mounted with `lumount`.
- x86 only: **Starting with the Solaris 10 1/06 release**, you cannot delete a boot environment that contains the active GRUB menu. Use the `lumake` or `luupgrade` commands to reuse the boot environment. To determine which boot environment contains the active GRUB menu, see [Chapter 13, “Managing the Oracle Solaris Boot Archives \(Tasks\),” in *Oracle Solaris Administration: Basic Administration*](#).

The syntax for the `ludelete` command is as follows:

```
# ludelete BE-name
```

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

EXAMPLE 7-3 Deleting an Inactive Boot Environment

In this example, the boot environment `second_disk` is deleted.

```
# ludelete second_disk
```

Displaying the Name of the Active Boot Environment

Use 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 122](#) to determine a boot environment’s status.

EXAMPLE 7-4 Displaying the Name of the Active Boot Environment

In this example, the name of the current boot environment is displayed.

```
# /usr/sbin/lucurr  
solaris10
```

Changing the Name of a Boot Environment

Renaming a boot environment is often useful when you upgrade the boot environment from one Oracle Solaris release to another release. For example, following an operating system upgrade, you might rename the boot environment `solaris8` to `solaris10`.

Use the `lurename` command to change the inactive boot environment’s name.

x86 only – Starting with the Solaris 10 1/06 release, the GRUB menu is automatically updated when you use the `Rename` menu or `lurename` command. The updated GRUB menu displays the boot environment’s name in the list of boot entries. For more information about the GRUB menu, see [“Booting Multiple Boot Environments” on page 56](#).

To find out how to determine the location of the GRUB menu’s `menu.lst` file, see [Chapter 13, “Managing the Oracle Solaris Boot Archives \(Tasks\)”](#), in *Oracle Solaris Administration: Basic Administration*.

Note the following limitations for naming a boot environment:

- The name must not exceed 30 characters in length.
- The name can consist only of alphanumeric characters and other ASCII characters that are not special to the UNIX shell. For more information, see the “Quoting” section of the `sh(1)` man page.
- The name can contain only single-byte, 8-bit characters.
- The name must be unique on the system.
- A boot environment must have the status `Complete` before you rename it. For more information, see [“Displaying the Status of All Boot Environments” on page 122](#) to determine a boot environment's status.
- You cannot rename a boot environment that has file systems mounted with `lumount` or `mount`.

The syntax of the `lurename` command is as follows:

```
# lurename -e BE-name -n new-name
-e BE-name      Specifies the inactive boot environment name to be changed
-n new-name     Specifies the new name of the inactive boot environment
```

In this example, `second_disk` is renamed to `third_disk`.

```
# lurename -e second_disk -n third_disk
```

Adding or Changing a Description Associated With a Boot Environment Name

You can associate a description with a boot environment name. The description never replaces the name. Although a boot environment name is restricted in length and characters, the description can be of any length and of any content. The description can be simple text or as complex as a `gif` file. You can create this description at these times:

- When you create a boot environment with the `luc create` command and use the `-A` option. For more information, see [“How to Create a Boot Environment for the First Time” on page 65](#).
- After the boot environment has been created by using the `ludesc` command. For more information, see the `ludesc(1M)` man page.

The syntax for the `ludesc` command is as follows:

```
# /usr/sbin/ludesc -n BE-name 'BE-description' -f filename
-n BE-name      Specifies the boot environment name.
```

BE-description Specifies the new description to be associated with the name.

filename Specifies the file to be associated with a boot environment name.

EXAMPLE 7-5 Adding a Description to a Boot Environment Name With Text

In this example, a boot environment description is added to a boot environment that is named `second_disk`. The description is text that is enclosed in single quotes.

```
# /usr/sbin/ludesc -n second_disk 'Oracle Solaris 10 8/11 test build'
```

EXAMPLE 7-6 Adding a Description to a Boot Environment Name With a File

In this example, a boot environment description is added to a boot environment that is named `second_disk`. The description is contained in a gif file.

```
# /usr/sbin/ludesc -n second_disk -f rose.gif
```

EXAMPLE 7-7 Displaying a Boot Environment Name From a Description

In this example, the name of the boot environment, `second_disk`, is displayed by using the `-A` option with the description.

```
# /usr/sbin/ludesc -A 'Oracle Solaris 10 8/11 test build'
second_disk
```

EXAMPLE 7-8 Displaying a Boot Environment Name From a Description in a File

In this example, the name of the boot environment, `second_disk`, is displayed by using the `-f` option and the name of the file that contains the description.

```
# /usr/sbin/ludesc -f rose.gif
second_disk
```

EXAMPLE 7-9 Displaying a Boot Environment Description From a Name

In this example, the description is displayed by using the `-n` option with the boot environment name.

```
# /usr/sbin/ludesc -n second_disk
Oracle Solaris 10 8/11 test build
```

Viewing the Configuration of a Boot Environment

Use 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.

The syntax for the `lufslist` command is as follows:

`lufslist -n BE-name`

BE-name Specifies the name of the boot environment to view file system specifics

The following example displays a list.

Filesystem	fstype	size(MB)	Mounted on
/dev/dsk/c0t0d0s1	swap	512.11	-
/dev/dsk/c0t4d0s3	ufs	3738.29	/
/dev/dsk/c0t4d0s4	ufs	510.24	/opt

Note – For an example of a list that contains non-global zones, see [“To View the Configuration of a Boot Environment's Non-Global Zone File Systems”](#) on page 144.

Upgrading the Oracle Solaris OS on a System With Non-Global Zones Installed

This chapter describes using Live Upgrade to upgrade a system that has non-global zones installed.

Note – This chapter describes Live Upgrade for UFS file systems. For procedures for migrating a UFS file system with non-global zones to a ZFS root pool, see [Chapter 13, “Live Upgrade for ZFS With Non-Global Zones Installed.”](#)

This chapter contains the following sections:

- For a summary of changes when you upgrade a system that contains non-global zones with Live Upgrade, see [“Upgrading With Live Upgrade and Installed Non-Global Zones \(Overview\)”](#) on page 131.
- For step-by-step instructions for creating a new boot environment and upgrading a system with Live Upgrade, see [“Creating and Upgrading a Boot Environment When Non-Global Zones Are Installed \(Tasks\)”](#) on page 137.
- For an example with brief instructions for creating a new boot environment and upgrading a system with Live Upgrade, see [“Upgrading a System With Non-Global Zones Installed \(Example\)”](#) on page 142.
- For general information about creating non-global zones, see *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*.

Upgrading With Live Upgrade and Installed Non-Global Zones (Overview)

Starting with the Solaris Solaris 10 8/07 release, you can upgrade or patch a system that contains non-global zones with Live Upgrade. If you have a system that contains non-global zones, Live Upgrade is the recommended program to upgrade and to add patches. Other upgrade programs

might require extensive upgrade time, because the time required to complete the upgrade increases linearly with the number of installed non-global zones. If you are patching a system with Live Upgrade, you do not have to take the system to single-user mode and you can maximize your system's uptime. The following list summarizes changes to accommodate systems that have non-global zones installed.

- A new package, `SUNWlucfg`, is required to be installed with the other Live Upgrade packages, `SUNWlur` and `SUNWluu`. This package is required for any system, not just a system with non-global zones installed.
- Creating a new boot environment from the currently running boot environment remains the same as in previous releases with one exception. You can specify a destination disk slice for a shared file system within a non-global zone. For more information, see [“Creating and Upgrading a Boot Environment When Non-Global Zones Are Installed \(Tasks\)”](#) on page 137.
- The `lumount` command now provides non-global zones with access to their corresponding file systems that exist on inactive boot environments. When the global zone administrator uses the `lumount` command to mount an inactive boot environment, the boot environment is mounted for non-global zones as well. See [“Using the lumount Command on a System That Contains Non-Global Zones”](#) on page 146.
- Comparing boot environments is enhanced. The `lucompare` command now generates a comparison of boot environments that includes the contents of any non-global zone. See [“To Compare Boot Environments for a System With Non-Global Zones Installed”](#) on page 145.
- Listing file systems with the `lufslst` command is enhanced to list file systems for both the global zone and the non-global zones. See [“To View the Configuration of a Boot Environment's Non-Global Zone File Systems”](#) on page 144.

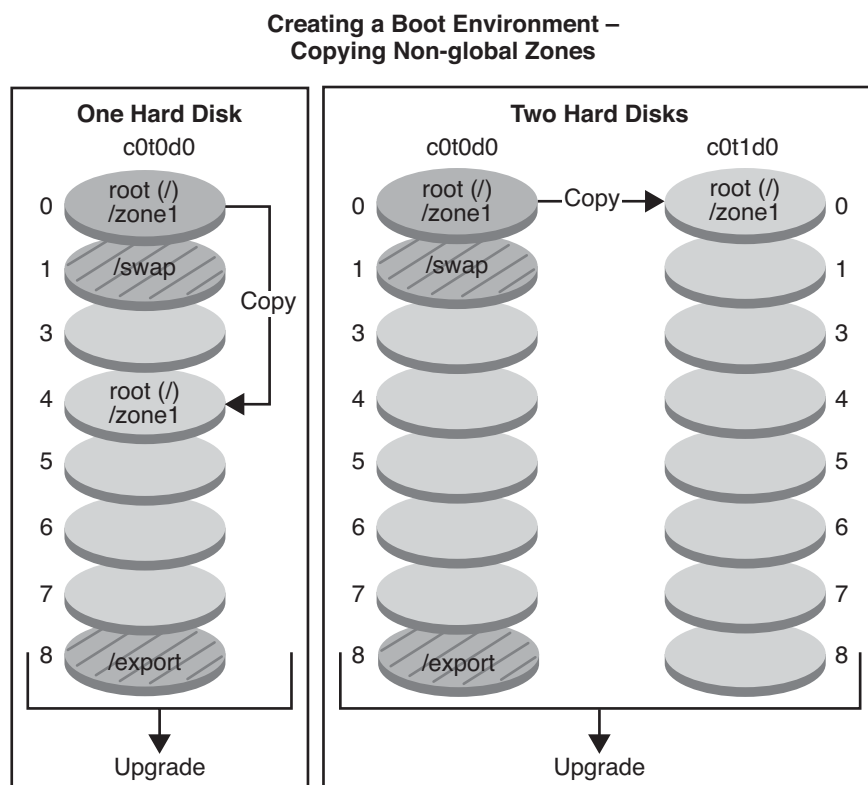
Understanding Oracle Solaris Zones and Live Upgrade

The Oracle Solaris Zones partitioning technology is used to virtualize operating system services and provide an isolated and secure environment for running applications. A non-global zone is a virtualized operating system environment created within a single instance of the Oracle Solaris OS, the global zone. When you create a non-global zone, you produce an application execution environment in which processes are isolated from the rest of the system.

Live Upgrade is a mechanism to copy the currently running system onto new slices. When non-global zones are installed, they can be copied to the inactive boot environment along with the global zone's file systems.

The following figure shows a non-global zone that is copied to the inactive boot environment along with the global zone's file system.

FIGURE 8-1 Creating a Boot Environment – Copying Non-Global Zones






Single disk command:

```
# lucreate -c bootenv1 \
-m /dev/dsk/c0t0d0s4:ufs \
-n bootenv2
```

Two disks command:

```
# lucreate -c bootenv1 \
-m /dev/dsk/c0t1d0s0:ufs \
-n bootenv2
```

-  Current release X
Critical file system root (/)
-  Inactive release X
Critical file systems root (/)
-  Shared file systems

- In this example of a system with a single disk, the root (/) file system is copied to c0t0d0s4. All non-global zones that are associated with the file system are also copied to s4. The /export file system and /swap volume are shared between the current boot environment, bootenv1, and the inactive boot environment, bootenv2. The lucreate command is as follows:

```
# lucreate -c bootenv1 -m /dev/dsk/c0t0d0s4:ufs -n bootenv2
```

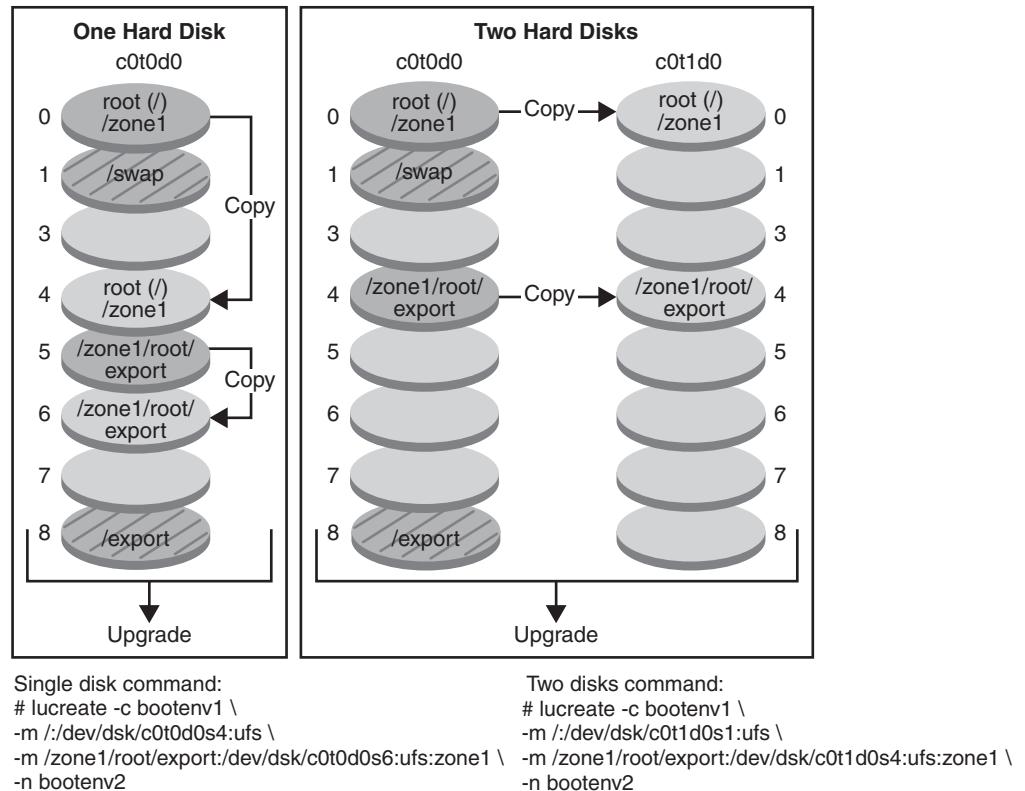
- In this example of a system with two disks, the root (/) file system is copied to `c0t1d0s0`. All non-global zones that are associated with the file system are also copied to `s0`. The `/export` file system and `/swap` volume are shared between the current boot environment, `bootenv1`, and the inactive boot environment, `bootenv2`. The `lucreate` command is as follows:

```
# lucreate -c bootenv1 -m /:/dev/dsk/c0t1d0s0:ufs -n bootenv2
```

The following figure shows that a non-global zone is copied to the inactive boot environment.

FIGURE 8-2 Creating a Boot Environment – Copying a Shared File System From a Non-Global Zone

Creating a Boot Environment – Non-global Zones Copying a Shared File System



- ☒ Current release X
Critical file system root (/)
- ☐ Inactive release X
Critical file systems root (/)
- ☒ Shared file systems

- In this example of a system with a single disk, the root (/) file system is copied to c0t0d0s4. All non-global zones that are associated with the file system are also copied to s4. The non-global zone, zone1, has a separate file system that was created by the zonecfg add fs command. The zone path is /zone1/root/export. To prevent this file system from being shared by the inactive boot environment, the file system is placed on a separate slice, c0t0d0s6. The /export file system and /swap volume are shared between the current boot environment, bootenv1, and the inactive boot environment, bootenv2. The lucreate command is as follows:

```
# lucreate -c bootenv1 -m /:/dev/dsk/c0t0d0s4:ufs \
-m /zone1/root/export:/dev/dsk/c0t0d0s6:ufs:zone1 -n bootenv2
```

- In this example of a system with two disks, the root (/) file system is copied to c0t1d0s0. All non-global zones that are associated with the file system are also copied to s0. The non-global zone, zone1, has a separate file system that was created by the zonecfg add fs command. The zone path is /zone1/root/export. To prevent this file system from being shared by the inactive boot environment, the file system is placed on a separate slice, c0t1d0s4. The /export file system and /swap volume are shared between the current boot environment, bootenv1, and the inactive boot environment, bootenv2. The lucreate command is as follows:

```
# lucreate -c bootenv1 -m /:/dev/dsk/c0t1d0s0:ufs \
-m /zone1/root/export:/dev/dsk/c0t1d0s4:ufs:zone1 -n bootenv2
```

Guidelines for Using Live Upgrade With Non-Global Zones (Planning)

Consider these issues when using Live Upgrade on a system with zones installed. It is critical to avoid zone state transitions during lucreate and lumount operations.

- When you use the lucreate command to create an inactive boot environment, if a given non-global zone is not running, then the zone cannot be booted until the lucreate operation has completed.
- When you use the lucreate command to create an inactive boot environment if a given non-global zone is running, the zone should not be halted or rebooted until the lucreate operation has completed.
- When an inactive boot environment is mounted with the lumount command, you cannot boot non-global zones or reboot them, although zones that were running before the lumount operation can continue to run.
- Because a non-global zone can be controlled by a non-global zone administrator as well as by the global zone administrator, to prevent any interaction, halt all zones during lucreate or lumount operations.

Problems can occur when the global zone administrator does not notify the non-global zone administrator of an upgrade with Live Upgrade.

When Live Upgrade operations are underway, non-global zone administrator involvement is critical. The upgrade affects the work of the administrators, who will be addressing the changes that occur as a result of the upgrade. Zone administrators should ensure that any local packages are stable throughout the sequence, handle any post-upgrade tasks such as configuration file adjustments, and generally schedule around the system outage.

For example, if a non-global zone administrator adds a package while the global zone administrator is copying the file systems with the lucreate command, the new package is not copied with the file systems and the non-global zone administrator is unaware of the problem.

Creating a Boot Environment When a Non-Global Zone Is on a Separate File System

Creating a new boot environment from the currently running boot environment remains the same as in previous releases with one exception. You can specify a destination disk slice for a shared file system within a non-global zone. This exception occurs under the following conditions:

- If on the current boot environment the `zonecfg add fs` command was used to create a separate file system for a non-global zone
- If this separate file system resides on a shared file system, such as `/zone/root/export`

To prevent this separate file system from being shared in the new boot environment, the `lucreate` command enables specifying a destination slice for a separate file system for a non-global zone. The argument to the `-m` option has a new optional field, *zonename*. This new field places the non-global zone's separate file system on a separate slice in the new boot environment. For more information about setting up a non-global zone with a separate file system, see the [zonecfg\(1M\)](#) man page.

Note – By default, any file system other than the critical file systems (`root (/)`, `/usr`, and `/opt` file systems) is shared between the current and new boot environments. Updating shared files in the active boot environment also updates data in the inactive boot environment. For example, the `/export` file system is a shared file system. If you use the `-m` option and the *zonename* option, the non-global zone's file system is copied to a separate slice and data is not shared. This option prevents non-global zone file systems that were created with the `zonecfg add fs` command from being shared between the boot environments.

Creating and Upgrading a Boot Environment When Non-Global Zones Are Installed (Tasks)

The following sections provide step-by-step procedures for upgrading when non-global zones are installed.

- [“Upgrading With Live Upgrade When Non-Global Zones Are Installed on a System \(Tasks\)” on page 138](#)

For an example with abbreviated steps, see [“Upgrading a System With Non-Global Zones Installed \(Example\)” on page 142](#).

▼ Upgrading With Live Upgrade When Non-Global Zones Are Installed on a System (Tasks)

The following procedure provides detailed instructions for upgrading with Live Upgrade for a system with non-global zones installed.

- 1 **Before running Live Upgrade for the first time, you must install the latest Live Upgrade packages from installation media and install the patches listed in the My Oracle Support knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on the [My Oracle Support](#) web site.**

The latest packages and patches ensure that you have all the latest bug fixes and new features in the release. Ensure that you install all the patches that are relevant to your system before proceeding to create a new boot environment.

The following substeps describe the steps in the My Oracle Support knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

- a. **Become superuser or assume an equivalent role.**

Note – Roles contain authorizations and privileged commands. For more information about roles, see “[Configuring RBAC \(Task Map\)](#)” in *System Administration Guide: Security Services*.

- b. **From the My Oracle Support web site, follow the instructions in the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) to remove and add Live Upgrade packages.**

The following instructions summarize the knowledge document steps for removing and adding the packages.

- Remove existing Live Upgrade packages.

The three Live Upgrade packages, SUNWluu, SUNWlur, and SUNWlucfg, comprise the software needed to upgrade or patch by using Live Upgrade. These packages include existing software, new features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading or patching to the target release fails. The SUNWlucfg package is new **starting with the Solaris 10 8/07 release**. If you are using Live Upgrade packages from a release previous to Solaris 10 8/07, you do not need to remove this package.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

- Install the new Live Upgrade packages.

You can install the packages by using the `liveupgrade20` command that is on the installation DVD or CD. The `liveupgrade20` command requires Java software. If your system does not have Java software installed, then you need to use the `pkgadd` command to install the packages. See the My Oracle Support knowledge document for more information.

- If you are using the Oracle Solaris Operating System DVD, change directories and run the installer:
 - Change directories.

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

Note – For SPARC based systems, the path to the installer is different for releases previous to the Solaris 10 10/08 release:

```
# cd /cdrom/cdrom0/s0/Solaris_10/Tools/Installers
```

- Run the installer

```
# ./liveupgrade20
```

The Oracle Solaris installation program GUI is displayed. If you are using a script, you can prevent the GUI from displaying by using the `-noconsole` and `-nodisplay` options.

- If you are using the Oracle Solaris Software – 2 CD, you can run the installer without changing the path.

```
% ./installer
```

- Verify that the packages have been installed successfully.

```
# pkgchk -v SUNWlucfg SUNWlur SUNWluu
```

c. If you are storing the patches on a local disk, create a directory such as `/var/tmp/lupatches`.

d. From the My Oracle Support web site, obtain the list of patches.

e. Change to the patch directory as in this example.

```
# cd /var/tmp/lupatches
```

f. Install the patches.

```
# patchadd -M path-to-patchespatch-id patch-id
```

path-to-patches is the path to the patches directory, such as `/var/tmp/lupatches`. *patch-id* is the patch number or numbers. Separate multiple patch names with a space.

Note – The patches need to be applied in the order specified in knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

g. Reboot the system if necessary. Certain patches require a reboot to be effective.

x86 only: Rebooting the system is required. Otherwise, Live Upgrade fails.

init 6

You now have the packages and patches necessary for a successful creation of a new boot environment.

2 Create the new boot environment.

```
# lucreate [-A 'BE_description'] [-c BE_name] \
-m mountpoint:device[,metadevice]:fs_options[:zonename] [-m ...] -n BE_name
-n BE_name
```

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

-A 'BE_description'

(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (*BE_name*). The description can be any length and can contain any characters.

-c *BE_name*

Assigns the name *BE_name* to the active boot environment. This option is not required and is only used when the first boot environment is created. If you run `lucreate` for the first time and you omit the `-c` option, the software creates a default name for you.

-m *mountpoint:device[,metadevice]:fs_options[:zonename]* [-m ...]

Specifies the file systems' configuration of the new boot environment in the `vfstab`. The file systems that are specified as arguments to `-m` can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.

- *mountpoint* can be any valid mount point or `-` (hyphen), indicating a swap partition.
- *device* field can be one of the following:
 - The name of a disk device, of the form `/dev/dsk/cwt:xdysz`
 - The name of a Solaris Volume Manager volume, of the form `/dev/md/dsk/dnum`
 - The name of a Veritas Volume Manager volume, of the form `/dev/md/vxfs/dsk/dnum`
 - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
- *fs_options* field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.

- swap, which indicates a swap volume. The swap mount point must be a – (hyphen).
- For file systems that are logical devices (mirrors), several keywords specify actions to be applied to the file systems. These keywords can create a logical device, change the configuration of a logical device, or delete a logical device. For a description of these keywords, see [“How to Create a Boot Environment With RAID-1 Volumes \(Mirrors\)” on page 78](#).
- *zonename* specifies that a non-global zone's separate file system be placed on a separate slice. This option is used when the zone's separate file system is in a shared file system such as `/zone1/root/export`. This option copies the zone's separate file system to a new slice and prevents this file system from being shared. The separate file system was created with the `zonecfg add fs` command.

In the following example, a new boot environment named `newbe` is created. The root (`/`) file system is placed on `c0t1d0s4`. All non-global zones in the current boot environment are copied to the new boot environment. The non-global zone named `zone1` is given a separate mount point on `c0t1d0s1`.

Note – By default, any file system other than the critical file systems (root (`/`), `/usr`, and `/opt` file systems) is shared between the current and new boot environments. The `/export` file system is a shared file system. If you use the `-m` option, the non-global zone's file system is placed on a separate slice and data is not shared. This option prevents zone file systems that were created with the `zonecfg add fs` command from being shared between the boot environments. See [`zonecfg\(1M\)`](#) for details.

```
# lucreate -n newbe -m /:/dev/dsk/c0t1d0s4:ufs -m /export:/dev/dsk/c0t1d0s1:ufs:zone1
```

3 Upgrade the boot environment.

The operating system image to be used for the upgrade is taken from the network.

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

-u	Upgrades an operating system image on a boot environment
-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

In this example, the new boot environment, `newbe`, is upgraded from a network installation image.

```
# luupgrade -n newbe -u -s /net/server/export/Solaris_10/combined.solaris_wos
```

4 (Optional) Verify that the boot environment is bootable.

The `lustatus` command reports if the boot environment creation is complete and bootable.

```
# lustatus
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
c0t1d0s0	yes	yes	yes	no	-
newbe	yes	no	no	yes	-

5 Activate the new boot environment.

```
# luactivate BE_name
```

BE_name specifies the name of the boot environment that is to be activated.

Note – For an x86 based system, the `luactivate` command is required when booting a boot environment for the first time. Subsequent activations can be made by selecting the boot environment from the GRUB, a feature of Oracle Solaris, menu. For step-by-step instructions, see [“x86: Activating a Boot Environment With the GRUB Menu” on page 108](#).

To successfully activate a boot environment, that boot environment must meet several conditions. For more information, see [“Activating a Boot Environment” on page 104](#).

6 Reboot.

```
# init 6
```



Caution – Use only the `init` or `shutdown` commands to reboot. If you use the `reboot`, `halt`, or `uadmin` commands, the system does not switch boot environments. The most recently active boot environment is booted again.

The boot environments have switched and the new boot environment is now the current boot environment.

7 (Optional) Fall back to a different boot environment.

If the new boot environment is not viable or you want to switch to another boot environment, see [Chapter 6, “Failure Recovery: Falling Back to the Original Boot Environment \(Tasks\)”](#).

Upgrading a System With Non-Global Zones Installed (Example)

The following procedure provides an example with abbreviated instructions for upgrading with Live Upgrade.

For detailed explanations of steps, see “[Upgrading With Live Upgrade When Non-Global Zones Are Installed on a System \(Tasks\)](#)” on page 138.

Upgrading With Live Upgrade When Non-Global Zones Are Installed on a System

The following example provides abbreviated descriptions of the steps to upgrade a system with non-global zones installed. In this example, a new boot environment is created by using the `lucreate` command on a system that is running the Oracle Solaris 10 release. This system has non-global zones installed and has a non-global zone with a separate file system on a shared file system, `zone1/root/export`. The new boot environment is upgraded to the Oracle Solaris 10 8/11 release by using the `luupgrade` command. The upgraded boot environment is activated by using the `luactivate` command.

Note – This procedure assumes that the system is running *Volume Manager*. For detailed information about managing removable media with the Volume Manager, refer to [System Administration Guide: Devices and File Systems](#).

1. Install required patches.

Ensure that you have the most recently updated patch list by consulting <http://support.oracle.com> (My Oracle Support). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support. In this example, `/net/server/export/patches` is the path to the patches.

```
# patchadd /net/server/export/patches
# init 6
```

2. Remove the Live Upgrade packages from the current boot environment.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

3. Insert the Oracle Solaris DVD or CD. Then install the replacement Live Upgrade packages from the target release.

```
# pkgadd -d /cdrom/cdrom0/Solaris_10/Product SUNWlucfg SUNWlur SUNWluu
```

4. Create a boot environment.

In the following example, a new boot environment named `newbe` is created. The root (`/`) file system is placed on `c0t1d0s4`. All non-global zones in the current boot environment are copied to the new boot environment. A separate file system was created with the `zonecfg add fs` command for `zone1`. This separate file system `/zone/root/export` is placed on a separate file system, `c0t1d0s1`. This option prevents the separate file system from being shared between the current boot environment and the new boot environment.

```
# lucreate -n newbe -m /:/dev/dsk/c0t1d0s4:ufs -m /export:/dev/dsk/c0t1d0s1:ufs:zone1
```

5. Upgrade the new boot environment.

In this example, `/net/server/export/Solaris_10/combined.solaris_wos` is the path to the network installation image.

```
# luupgrade -n newbe -u -s /net/server/export/Solaris_10/combined.solaris_wos
```

6. (Optional) Verify that the boot environment is bootable.

The `lustatus` command reports if the boot environment creation is complete.

```
# lustatus
boot environment      Is      Active  Active  Can      Copy
Name                  Complete Now    OnReboot Delete   Status
-----
c0t1d0s0              yes     yes     yes     no       -
newbe                  yes     no      no      yes      -
```

7. Activate the new boot environment.

```
# luactivate newbe
# init 6
```

The boot environment `newbe` is now active.

8. (Optional) Fall back to a different boot environment. If the new boot environment is not viable or you want to switch to another boot environment, see [Chapter 6, “Failure Recovery: Falling Back to the Original Boot Environment \(Tasks\)”](#).

Administering Boot Environments That Contain Non-Global Zones

The following sections provide information about administering boot environments that contain non-global zones.

▼ To View the Configuration of a Boot Environment's Non-Global Zone File Systems

Use this procedure to display a list of file systems for both the global zone and the non-global zones.

- Display the list of file systems.

```
# lufslist -n BE_name
BE_name    Specifies the name of the boot environment to view file system specifics
```

Example 8–1 List File Systems With Non-Global Zones

The following example displays a list of file systems that include non-global zones.


```
# lufslist -n s3
boot environment name: s3
This boot environment is currently active.
This boot environment will be active on next system boot.
```

Filesystem	fstype	device size	Mounted on	Mount Options
/dev/dsk/c0t0d0s1	swap	2151776256	-	-
/dev/dsk/c0t0d0s3	ufs	10738040832	/	-
/dev/dsk/c0t0d0s7	ufs	10487955456	/export	-
zone <zone1> within boot environment <s3>				
/dev/dsk/c0t0d0s5	ufs	5116329984	/export	-

▼ To Compare Boot Environments for a System With Non-Global Zones Installed

The `lucompare` command now generates a comparison of boot environments that includes the contents of any non-global zone.

● Compare the current and new boot environments.

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

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

Example 8-2 Comparing Boot Environments

In this example, current 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
```

Using the `lumount` Command on a System That Contains Non-Global Zones

The `lumount` command provides non-global zones with access to their corresponding file systems that exist on inactive boot environments. When the global zone administrator uses the `lumount` command to mount an inactive boot environment, the boot environment is mounted for non-global zones as well.

In the following example, the appropriate file systems are mounted for the boot environment, `newbe`, on `/mnt` in the global zone. For non-global zones that are running, mounted, or ready, their corresponding file systems within `newbe` are also made available on `/mnt` within each zone.

```
# lumount -n newbe /mnt
```

For more information about mounting, see the [lumount\(1M\)](#) man page.

Live Upgrade Examples

This chapter provides examples of creating a boot environment, then upgrading and activating the new boot environment which then becomes the currently running system.

Note – This chapter describes Live Upgrade for UFS file systems. For procedures for migrating a UFS file system to a ZFS root pool or creating and installing a ZFS root pool, see [Chapter 12](#), “Creating a Boot Environment for ZFS Root Pools.”

This chapter contains the following sections:

- “Example of Upgrading With Live Upgrade” on page 147
- “Example of Detaching and Upgrading One Side of a RAID-1 Volume (Mirror)” on page 156
- “Example of Migrating From an Existing Volume to a Solaris Volume Manager RAID-1 Volume” on page 159
- “Example of Creating an Empty Boot Environment and Installing a Flash Archive” on page 159

Example of Upgrading With Live Upgrade

In this example, a new boot environment is created by using the `lucreate` command on a system that is running the Solaris 9 release. The new boot environment is upgraded to the Oracle Solaris 10 8/11 release by using the `luupgrade` command. The upgraded boot environment is activated by using the `luactivate` command. An example of falling back to the original boot environment is also given.

Preparing to Use Live Upgrade

Before running Live Upgrade for the first time, you must install the latest Live Upgrade packages from installation media and install the patches listed in the My Oracle Support

knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on the [My Oracle Support](#) web site.

The latest packages and patches ensure that you have all the latest bug fixes and new features in the release. Ensure that you install all the patches that are relevant to your system before proceeding to create a new boot environment.

The following steps describe the steps in the My Oracle Support knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

Note – This procedure assumes that the system is running *Volume Manager*. For detailed information about managing removable media with the Volume Manager, refer to [System Administration Guide: Devices and File Systems](#).

1. Become superuser or assume an equivalent role.

Note – Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in [System Administration Guide: Security Services](#).

2. From the My Oracle Support web site, follow the instructions in the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) to remove and add Live Upgrade packages.

- a. Remove existing Live Upgrade packages.

The three Live Upgrade packages, SUNWluu, SUNWlur, and SUNWlucfg, comprise the software needed to upgrade or patch by using Live Upgrade. These packages include existing software, new features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading or patching to the target release fails. The SUNWlucfg package is new **starting with the Solaris 10 8/07 release**. If you are using Live Upgrade packages from a release previous to Oracle Solaris 10 8/07, you do not need to remove this package.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

- b. Install the new Live Upgrade packages.

You can install the packages by using the `liveupgrade20` command that is on the installation DVD or CD or by using the `pkgadd` command. The `liveupgrade20` command requires Java software. If your system does not have Java software installed, then you need to use the `pkgadd` command to install the packages. See the My Oracle Support knowledge document for more information.

- If you are using the Oracle Solaris Operating System DVD, change directories and run the installer:

- Change directories.

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

Note – For SPARC based systems, the path to the installer is different for releases previous to the Solaris 10 10/08 release:

```
# cd /cdrom/cdrom0/s0/Solaris_10/Tools/Installers
```

- Run the installer

```
# ./liveupgrade20 -noconsole -nodisplay
```

The `-noconsole` and `-nodisplay` options prevent the character user interface (CUI) from displaying.

Note – The Live Upgrade CUI is no longer supported.

- If you are using the Oracle Solaris Software – 2 CD, you can run the installer without changing the path.

```
% ./installer
```

- Verify that the packages have been installed successfully.

```
# pkgchk -v SUNWlucfg SUNWlur SUNWluu
```

3. Install the patches listed in knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

- a. If you are storing the patches on a local disk, create a directory such as `/var/tmp/lupatches`.
- b. From the My Oracle Support web site, obtain the list of patches.
- c. Change to the patch directory as in this example.

```
# cd /var/tmp/lupatches
```

- d. Install the patches.

```
# patchadd -M path-to-patchespatch-id patch-id
```

path-to-patches is the path to the patch directory such as `/var/tmp/lupatches`.

patch-id is the patch number or numbers. Separate multiple patch names with a space.

Note – The patches need to be applied in the order specified in knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

- e. Reboot the system if necessary. Certain patches require a reboot to be effective.

x86 only: Rebooting the system is required. Otherwise, Live Upgrade fails.

init 6

You now have the packages and patches necessary for a successful creation of a new boot environment.

Using Pre Flight Checker Tool

You can use the Live Upgrade Pre Flight Check tool to detect system configuration issues that would likely cause a Live Upgrade operation to fail. This tool needs to be run before invoking the `lucreate` command to create a boot environment.

The following table describes how to use the Pre Flight Checker tool to run the system check and migrate from one filesystem to another.

Task	Sample Usage	Sample Output
To run system check	<code># lupc -s</code>	This system has Patch level/IDR of IDR147654-04. It is advisable to apply the latest Live Upgrade patch from MOS.
To migrate from one UFS filesystem to another	<code># lupc --ufs2ufs -m /:/dev/dsk/c1t5d0s0:ufs</code>	This system has Patch level/IDR of 121430-75 It is advisable to apply the latest Live Upgrad patch from MOS. File system has sufficient space to accommodate ABE Required space is 4995481 KB and fs has 4131006 KB
To migrate from one ZFS filesystem to another	<code># lupc -zfs2zfs -p newpool</code>	This system has Patch level/IDR of IDR147654-04 It is advisable to apply the latest Live Upgradepatch from MOS. File system has sufficient space to accommodate ABE Required space is 20787.2 KB and fs has 10422845.44 KB
To migrate from one UFS filesystem to ZFS filesystem	<code># lupc --ufs2zfs -p bepool -D /var</code>	This system has Patch level/IDR of 121430-75 It is advisable to apply the latest Live Upgrade patch from MOS. File system has sufficient space to accommodate ABE Required space is 4995481 KB and fs has 20761804.8 KB File system has sufficient space to accommodate ABE Required space is 5610220 KB and fs has 20761804.8 KB

To Create a Boot Environment

The source boot environment is named `c0t4d0s0` by using the `-c` option. Naming the source boot environment is required only when the first boot environment is created. For more information about naming using the `-c` option, see the description in “To Create a Boot Environment for the First Time” [Step 1](#).

The new boot environment is named `c0t15d0s0`. The `-A` option creates a description that is associated with the boot environment name.

The root (`/`) file system is copied to the new boot environment. Also, a new swap slice is created rather than sharing the source boot environment's swap slice.

```
# lucreate -A 'BE_description' -c /dev/dsk/c0t4d0s0 -m /:/dev/dsk/c0t15d0s0:ufs\
-m -:/dev/dsk/c0t15d0s1:swap -n /dev/dsk/c0t15d0s0
```

To Upgrade the Inactive Boot Environment

The inactive boot environment is named `c0t15d0s0`. The operating system image to be used for the upgrade is taken from the network.

```
# luupgrade -n c0t15d0s0 -u -s /net/ins-svr/export/Solaris_10 \
combined.solaris_wos
```

To Check if Boot Environment Is Bootable

The `lustatus` command reports if the boot environment creation is complete. `lustatus` also shows if the boot environment is bootable.

```
# lustatus
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
c0t4d0s0	yes	yes	yes	no	-
c0t15d0s0	yes	no	no	yes	-

To Activate the Inactive Boot Environment

The `c0t15d0s0` boot environment is made bootable with the `luactivate` command. The system is then rebooted and `c0t15d0s0` becomes the active boot environment. The `c0t4d0s0` boot environment is now inactive.

```
# luactivate c0t15d0s0
# init 6
```

(Optional) To Fall Back to the Source Boot Environment

The following procedures for falling back depend on your new boot environment activation situation:

- For SPARC based systems:
 - The activation is successful, but you want to return to the original boot environment. See [Example 9-1](#).
 - The activation fails and you can boot back to the original boot environment. See [Example 9-2](#).
 - The activation fails and you must boot back to the original boot environment by using media or a net installation image. See [Example 9-3](#).
- For x86 based systems, **starting with the Solaris 10 1/06 release** and when you use the GRUB, a feature of Oracle Solaris, menu:
 - The activation fails, the GRUB menu is displayed correctly, but the new boot environment is not bootable. See [Example 9-4](#)
 - The activation fails and the GRUB menu does not display. See [Example 9-5](#).

EXAMPLE 9-1 SPARC: To Fall Back Despite Successful Boot Environment Creation

In this example, the original `c0t4d0s0` boot environment is reinstated as the active boot environment although it was activated successfully. The device name is `first_disk`.

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

EXAMPLE 9-2 SPARC: To Fall Back From a Failed Boot Environment Activation

In this example, the new boot environment was not bootable. You must return to the OK prompt before booting from the original boot environment, `c0t4d0s0`, in single-user mode.

```
OK boot net -s
# /sbin/luactivate first_disk
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# init 6
```

The original boot environment, `c0t4d0s0`, becomes the active boot environment.

EXAMPLE 9-3 SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image

In this example, the new boot environment was not bootable. You cannot boot from the original boot environment and must use media or a net installation image. The device is `/dev/dsk/c0t4d0s0`. The original boot environment, `c0t4d0s0`, becomes the active boot environment.

```
OK boot net -s
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

EXAMPLE 9-4 x86: To Fall Back to the Original Boot Environment By Using the GRUB Menu

Starting with the Solaris 10 1/06 release, the following example provides the steps to fall back by using the GRUB menu.

In this example, the GRUB menu is displayed correctly, but the new boot environment is not bootable. To enable a fallback, the original boot environment is booted in single-user mode.

1. Become superuser or assume an equivalent role.

Note – Roles contain authorizations and privileged commands. For more information about roles, see “[Configuring RBAC \(Task Map\)](#)” in *System Administration Guide: Security Services*.

2. To display the GRUB menu, reboot the system.

```
# init 6
```

The GRUB menu is displayed.

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris                               |
|Solaris failsafe                      |
|second_disk                          |
|second_disk failsafe                  |
+-----+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

3. From the GRUB menu, select the original boot environment. The boot environment must have been created with GRUB software. A boot environment that was created before the **Solaris 10 1/06 release** is not a GRUB boot environment. If you do not have a bootable GRUB boot environment, then skip to [Example 9-5](#).

EXAMPLE 9-4 x86: To Fall Back to the Original Boot Environment By Using the GRUB Menu
(Continued)

4. Edit the GRUB menu by typing: **e**.
5. Select kernel `/boot/multiboot` by using the arrow keys and type **e**. The grub edit menu is displayed.

```
grub edit>kernel /boot/multiboot
```

6. Boot to single user mode, by typing **-s**.

```
grub edit>kernel /boot/multiboot -s
```

7. Boot and mount the boot environment. Then activate it.

```
# b
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

EXAMPLE 9-5 x86: To Fall Back to the Original Boot Environment With the GRUB Menu by Using the DVD or CD

Starting with the Solaris 10 1/06 release, the following example provides the steps to fall back by using the DVD or CD.

In this example, the new boot environment was not bootable. Also, the GRUB menu does not display. To enable a fallback, the original boot environment is booted in single-user mode.

1. Insert the Oracle Solaris Operating System for x86 Platforms DVD or Oracle Solaris Software for x86 Platforms - 1 CD.
2. Become superuser or assume an equivalent role.

Note – Roles contain authorizations and privileged commands. For more information about roles, see [“Configuring RBAC \(Task Map\)” in *System Administration Guide: Security Services*](#).

3. Boot from the DVD or CD.

```
# init 6
```

The GRUB menu is displayed.

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
```

```
+-----+
|Oracle Solaris 10 8/11                               |
|Oracle Solaris 10 8/11 Serial Console ttya           |
```

EXAMPLE 9-5 x86: To Fall Back to the Original Boot Environment With the GRUB Menu by Using the DVD or CD (Continued)

```
|Oracle Solaris 10 8/11 Serial Console ttyb (for lx50, v60x and v65x      |
+-----+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

4. Wait for the default option to boot or choose any option displayed.

The installation screen is displayed.

```
+-----+
|
|Select the type of installation you want to perform:
|
|      1 Solaris Interactive
|      2 Custom JumpStart
|      3 Solaris Interactive Text (Desktop session)
|      4 Solaris Interactive Text (Console session)
|      5 Apply driver updates
|      6 Single user shell
|
|      Enter the number of your choice followed by the <ENTER> key.
|      Alternatively, enter custom boot arguments directly.
|
|      If you wait 30 seconds without typing anything,
|      an interactive installation will be started.
|
+-----+
```

5. Choose the “Single user shell” option.

The following message is displayed.

```
Do you wish to automatically update the boot archive? y /n
```

6. Type: n

```
Starting shell...
#
```

You are now in single user mode.

7. Mount the boot environment. Then activate and reboot.

```
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

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

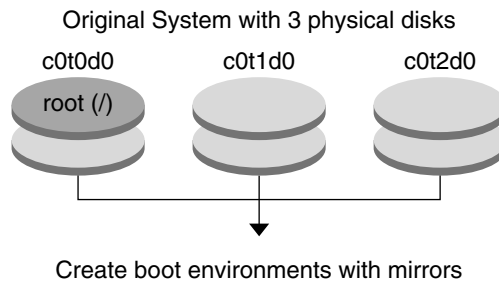
This example shows you how to do the following tasks:

- Create a RAID-1 volume (mirror) on a new boot environment
- Break the mirror and upgrade one half of the mirror
- Attach the other half of the mirror, the concatenation, to the new mirror

Figure 9–1 shows the current boot environment, which contains three physical disks.

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

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



1. Create a new boot environment, `second_disk`, that contains a mirror.

The following command performs these tasks.

- `lucreate` configures a UFS file system for the mount point `root (/)`. A mirror, `d10`, is created. This mirror is the receptacle for the current boot environment's `root (/)` file system, which is copied to the mirror `d10`. All data on the mirror `d10` is overwritten.
- Two slices, `c0t1d0s0` and `c0t2d0s0`, are specified to be used as submirrors. These two submirrors are attached to mirror `d10`.

```
# lucreate -c first_disk -n second_disk \
-m /dev/md/dsk/d10:ufs,mirror \
-m /dev/dsk/c0t1d0s0:attach \
-m /dev/dsk/c0t2d0s0:attach
```

2. Activate the `second_disk` boot environment.

```
# /sbin/luactivate second_disk
# init 6
```

3. Create another boot environment, `third_disk`.

The following command performs these tasks.

- `lucreate` configures a UFS file system for the mount point `root (/)`. A mirror, `d20`, is created.
- Slice `c0t1d0s0` is removed from its current mirror and is added to mirror `d20`. The contents of the submirror, the `root (/)` file system, are preserved and no copy occurs.

```
# lucreate -n third_disk \
-m /:/dev/md/dsk/d20:ufs,mirror \
-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

4. Upgrade the new boot environment, `third_disk`

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

5. Add a patch to the upgraded boot environment.

```
# luupgrade -t n third_disk -s /net/patches 222222-01
```

6. Activate the `third_disk` boot environment to make this boot environment the currently running system.

```
# /sbin/luactivate third_disk
# init 6
```

7. Delete the boot environment `second_disk`.

```
# ludelete second_disk
```

8. The following commands perform these tasks.

- Clear mirror `d10`.
- Check for the number for the concatenation of `c0t2d0s0`.
- Attach the concatenation that is found by the `metastat` command to the mirror `d20`. The `metattach` command synchronizes the newly attached concatenation with the concatenation in mirror `d20`. All data on the concatenation is overwritten.

```
# metaclear d10
# metastat -p | grep c0t2d0s0
dnum 1 1 c0t2d0s0
# metattach d20 dnum
```

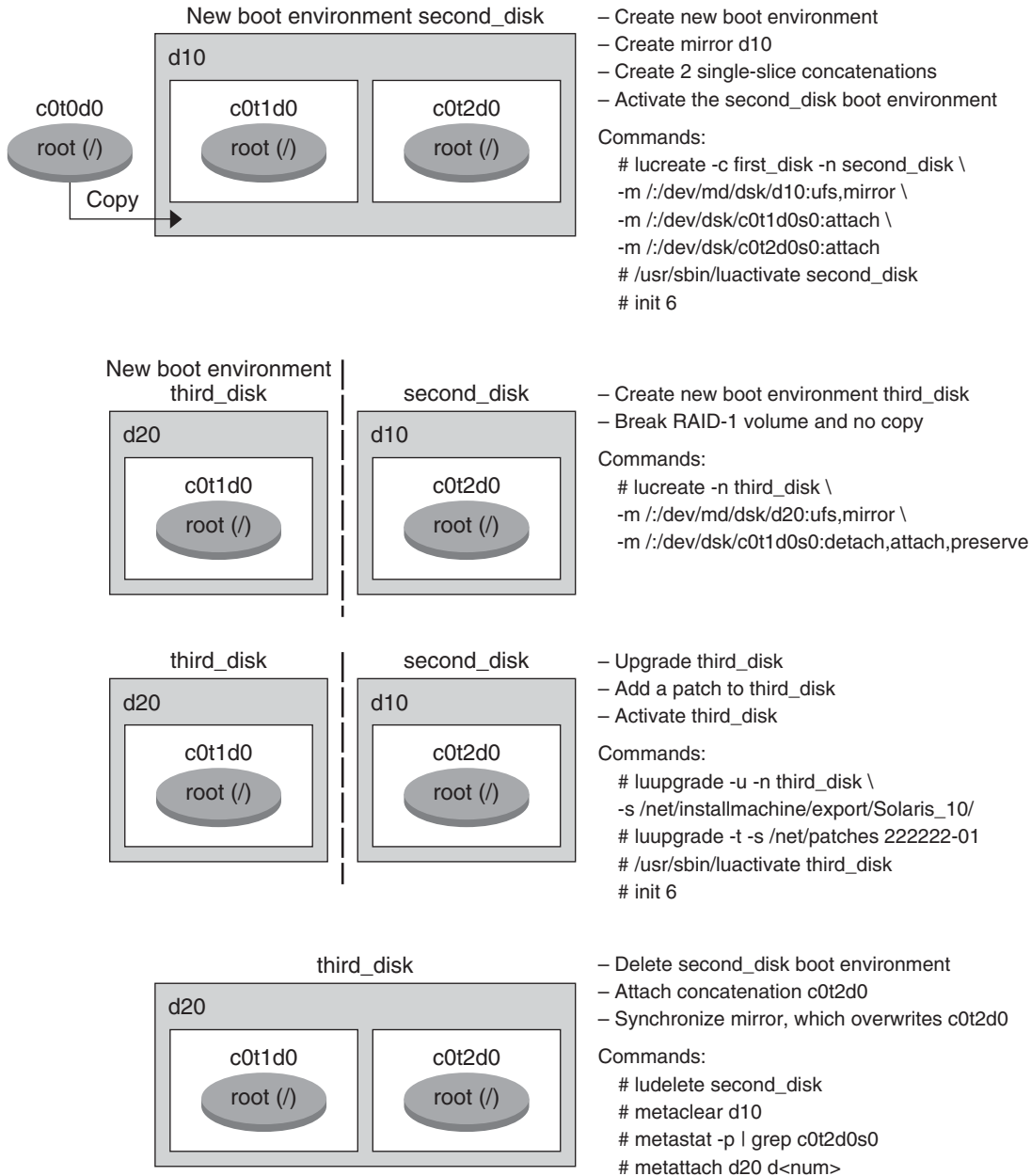
num Is the number found in the `metastat` command for the concatenation

The new boot environment, `third_disk`, has been upgraded and is the currently running system. `third_disk` contains the `root (/)` file system that is mirrored.

Figure 9–2 shows the entire process of detaching a mirror and upgrading the mirror by using the commands in the preceding example.

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

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



Example of Migrating From an Existing Volume to a Solaris Volume Manager RAID-1 Volume

Live Upgrade enables the creation of a new boot environment on RAID-1 volumes (mirrors). The current boot environment's file systems can be on any of the following:

- A physical storage device
- A Solaris Volume Manager controlled RAID-1 volume
- A Veritas VXFSS controlled volume

However, the new boot environment's target must be a Solaris Volume Manager RAID-1 volume. For example, the slice that is designated for a copy of the root (/) file system must be /dev/vx/dsk/rootvol. rootvol is the volume that contains the root (/) file system.

In this example, the current boot environment contains the root (/) file system on a volume that is not a Solaris Volume Manager volume. The new boot environment is created with the root (/) file system on the Solaris Volume Manager RAID-1 volume c0t2d0s0. The `lucreate` command migrates the current volume to the Solaris Volume Manager volume. The name of the new boot environment is `svm_be`. The `lustatus` command reports if the new boot environment is ready to be activated and be rebooted. The new boot environment is activated to become the current boot environment.

```
# lucreate -n svm_be -m /:/dev/md/dsk/d1:mirror,ufs \
-m /:/dev/dsk/c0t2d0s0:attach
# lustatus
# luactivate svm_be
# lustatus
# init 6
```

Example of Creating an Empty Boot Environment and Installing a Flash Archive

The following procedures cover the three-step process:

- Creating the empty boot environment
- Installing the archive
- Activating the boot environment which then becomes the currently running boot environment.

The `lucreate` command creates a boot environment that is based on the file systems in the active boot environment. When you use the `lucreate` command with the `-s` option, `lucreate` quickly creates an empty boot environment. The slices are reserved for the file systems specified, but no file systems are copied. The boot environment is named, but not actually

created until installed with a Flash Archive. When the empty boot environment is installed with an archive, file systems are installed on the reserved slices. The boot environment is then activated.

To Create an Empty Boot Environment

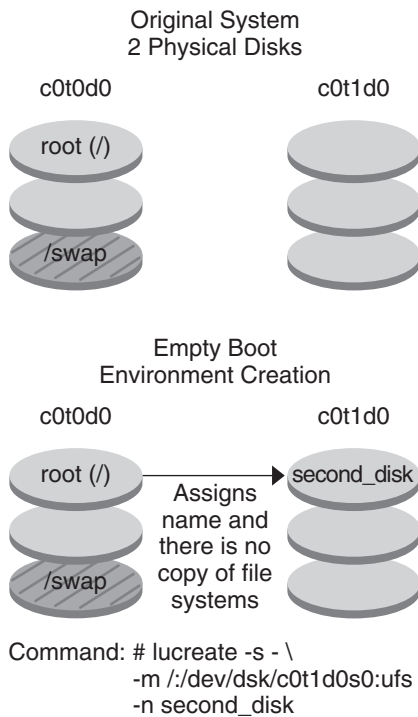
In this first step, an empty boot environment is created. Slices are reserved for the file systems that are specified, but no copy of file systems from the current boot environment occurs. The new boot environment is named `second_disk`.

```
# lucreate -s - -m /:/dev/dsk/c0t1d0s0:ufs \
-n second_disk
```

The boot environment is ready to be populated with a Flash Archive.

Figure 9–3 shows the creation of an empty boot environment.

FIGURE 9–3 Creating an Empty Boot Environment



To Install a Flash Archive on the New Boot Environment

In this second step, an archive is installed on the `second_disk` boot environment that was created in the previous example. The archive is located on the local system. The operating system versions for the `-s` and `-a` options are both Oracle Solaris 10 8/11 releases. The archive is named `Solaris_10.flar`.

```
# luupgrade -f -n second_disk \  
-s /net/installmachine/export/Solaris_10/OS_image \  
-a /net/server/archive/10.flar
```

The boot environment is ready to be activated.

To Activate the New Boot Environment

In this last step, the `second_disk` boot environment is made bootable with the `luactivate` command. The system is then rebooted and `second_disk` becomes the active boot environment.

```
# luactivate second_disk  
# init 6
```

- For step-by-step information about creating an empty boot environment, see [“How to Create an Empty Boot Environment for a Flash Archive”](#) on page 76.
- For step-by-step information about creating a Flash Archive, see Chapter 3, “Creating Flash Archives (Tasks),” in *Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation)*.
- For step-by-step information about activating a boot environment or falling back to the original boot environment, see Chapter 6, “Failure Recovery: Falling Back to the Original Boot Environment (Tasks).”

PART II

Upgrading and Migrating With Live Upgrade to a ZFS Root Pool

This part provides an overview and instructions for using Live Upgrade to create and upgrade an inactive boot environment on ZFS storage pools. Also, you can migrate your UFS root (/) file system to a ZFS root pool.

Live Upgrade and ZFS (Overview)

With Live Upgrade you can migrate your UFS file systems to a ZFS root pool and create ZFS root file systems from an existing ZFS root pool.

Note – Creating boot environments with Live Upgrade is new in the Solaris 10 10/08 release. When performing a Live Upgrade for a UFS file system, both the command-line parameters and operation of Live Upgrade remain unchanged. To find out how to perform a Live Upgrade on a system with UFS file systems, see [Part I, “Upgrading With Live Upgrade,”](#) of this book.

The following sections provide an overview of these tasks:

- [“Introduction to Using Live Upgrade With ZFS” on page 167.](#)
- Migrating a UFS file system with or without SVM volumes.
 - [“Migrating From a UFS File System to a ZFS Root Pool” on page 167](#)
 - [“Migrating a UFS File System With Solaris Volume Manager Volumes Configured to a ZFS Root File System” on page 171](#)
- Creating a new ZFS boot environment within the existing root pool or on another root pool.
 - [“Creating a New Boot Environment Within the Same Root Pool” on page 172](#)
 - [“Creating a New Boot Environment on Another Root Pool” on page 174](#)
- [“Creating a New Boot Environment From a Source Other Than the Currently Running System” on page 175.](#)
- [“Creating a ZFS Boot Environment on a System With Non-Global Zones Installed” on page 176.](#)

What's New in Oracle Solaris 10 8/11 Release

Starting with the Oracle Solaris 10 8/11 release, the ZFS file system has the following installation enhancements:

- You can use the Live Upgrade `luupgrade` command to install a flash archive to a ZFS rooted alternate boot environment.

Unlike the ZFS flash archives in the previous releases, a flash archive created on a ZFS root master system does not contain all the existing boot environments. Instead, the archive contains only the active ZFS boot environment. The archive does not include those datasets that are excluded explicitly with the `-D` option of the `lucreate` command and the user data present in the top-level pool dataset. The swap and dump volumes are not included into the archive but are created when the flash archive is installed.

For more information about ZFS flash archives creation and installation, refer to “[Installing a ZFS Root File System \(Oracle Solaris Flash Archive Installation\)](#)” in *Oracle Solaris ZFS Administration Guide*.

- You can use the `-D` option of the Live Upgrade `lucreate` command to create a separate dataset for `/var` when you migrate a UFS root file system to a ZFS root file system.

However, you cannot use the `-D` option to specify a separate dataset for any other OS critical file system. The `/var` file system is the only OS critical file system that is allowed to be a separate dataset in a ZFS root boot environment. The `-D` option is allowed only when the source boot environment has a UFS root and the target boot environment being created has a ZFS root.

For more information, refer to “[Migrating From a UFS Root \(/\) File System to ZFS Root Pool](#)” on page 168.

For detailed instructions and limitations, see [Chapter 4, “Installing and Booting an Oracle Solaris ZFS Root File System,”](#) in *Oracle Solaris ZFS Administration Guide*.

What's New in the Solaris 10 10/09 Release

Starting with the Solaris 10 10/09 release, you can set up a JumpStart profile to identify a flash archive of a ZFS root pool.

A flash archive can be created on a system that is running a UFS root file system or a ZFS root file system. A flash archive of a ZFS root pool contains the entire pool hierarchy except for the swap and dump volumes and any excluded datasets. The swap and dump volumes are created when the flash archive is installed.

You can use the flash archive installation method as follows:

- Generate a flash archive that can be used to install and boot a system with a ZFS root file system.
- Perform a JumpStart installation of a system by using a ZFS flash archive.

Note – Creating a ZFS flash archive backs up an entire root pool, not individual boot environments. Individual datasets within the pool can be excluded by using the `flarcreate` and `flar` command's `-D` option.

For detailed instructions and limitations, see “[Installing a ZFS Root File System \(Oracle Solaris Flash Archive Installation\)](#)” in *Oracle Solaris ZFS Administration Guide*.

Introduction to Using Live Upgrade With ZFS

If you have a UFS file system, Live Upgrade works the same as in previous releases. You can now migrate from UFS file systems to a ZFS root pool and create new boot environments within a ZFS root pool. For these tasks, the `lucreate` command has been enhanced with the `-p` option. The `lucreate` command has the following syntax:

```
# lucreate [-c active-BE-name] -n BE-name [-p zfs-root-pool [-D /var]]
```

The `-p` option specifies the ZFS pool in which a new boot environment resides. This option can be omitted if the source and target boot environments are within the same pool.

The `lucreate` command `-m` option is not supported with ZFS. Other `lucreate` command options work as usual, with some exceptions. For limitations, see “[System Requirements and Limitations When Using Live Upgrade](#)” on page 177.

For ZFS information, including overview, planning, and step-by-step instructions, see *Oracle Solaris ZFS Administration Guide*

Migrating From a UFS File System to a ZFS Root Pool

If you create a boot environment from the currently running system, the `lucreate` command copies the UFS root (`/`) file system to a ZFS root pool. The copy process might take time, depending on your system.

When you are migrating from a UFS file system, the source boot environment can be a UFS root (`/`) file system on a disk slice. You cannot create a boot environment on a UFS file system from a source boot environment on a ZFS root pool.

Migrating From a UFS Root (/) File System to ZFS Root Pool

The commands described in this section create a ZFS root pool and a new boot environment from a UFS root (/) file system in the ZFS root pool. A ZFS root pool must exist before the `lucreate` operation and must be created with slices rather than whole disks to be upgradeable and bootable. The disk must have an SMI label rather than an EFI label. For more limitations, see [“System Requirements and Limitations When Using Live Upgrade” on page 177](#).

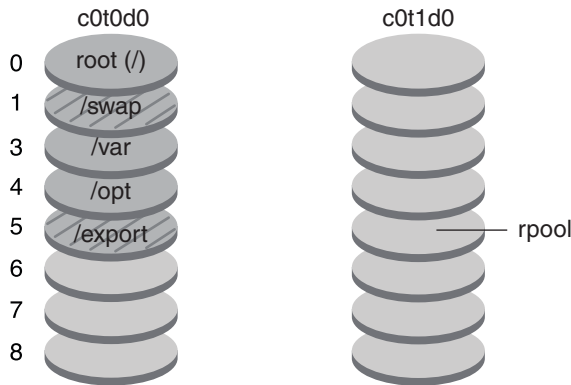
The following figure shows the `zpool` command that creates a root pool, `rpool` on a separate slice, `c0t1d0s5`. The disk slice `c0t0d0s0` contains a UFS root (/) file system. In the `lucreate` command, the `-c` option names the currently running system, `c0t0d0`, that is a UFS root (/) file system. The `-n` option assigns the name to the boot environment to be created, `new-zfsBE`. The `-p` option specifies where to place the new boot environment, `rpool`. The UFS `/export` file system and the `/swap` volume are not copied to the new boot environment.

FIGURE 10-1 Migrating From a UFS File System to a ZFS Root Pool

Migrating From a UFS File System to a ZFS Root Pool

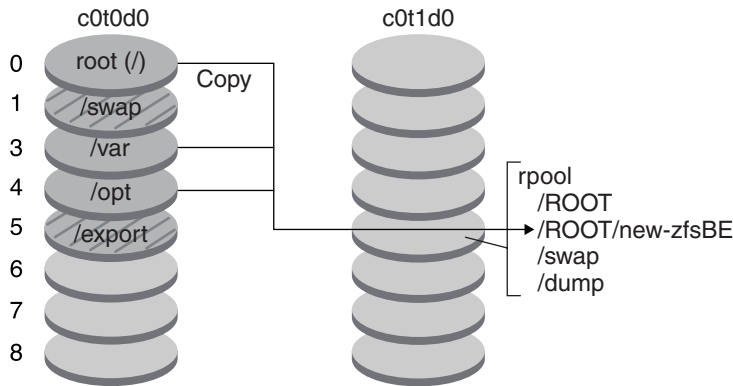
① Create a ZFS root pool on a slice.

zpool create rpool c0t1d0s5



② Copy the UFS boot environment to rpool.

lucreate -c c0t0d0 -n new-zfsBE -p rpool



To specify the creation of a separate dataset for `/var` in an alternate boot environment, use the `-D` option of the `lucreate` command.

```
lucreate -c c0t0d0 -n new-zfsBE -p rpool -D /var
```

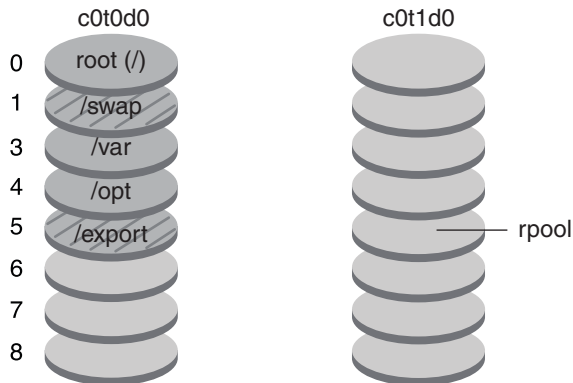
The following diagram shows the datasets created in `rpool` as a part of this sample `lucreate` command.

FIGURE 10-2 Migrating From a UFS File System to a ZFS Root Pool

Migrating From a UFS File System to a ZFS Root Pool

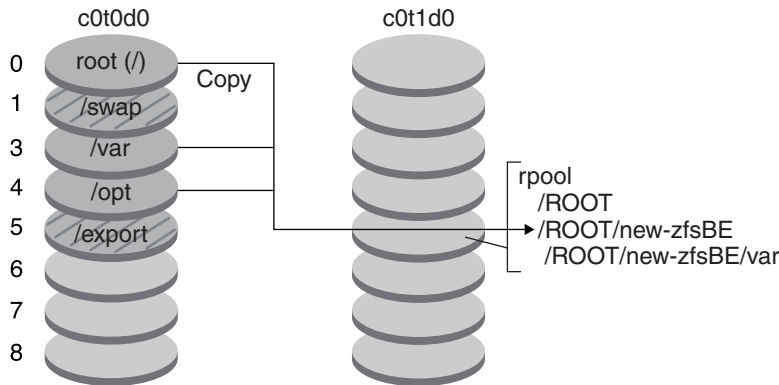
① Create a ZFS root pool on a slice.

zpool create rpool c0t1d0s5



② Copy the UFS boot environment to rpool.

lucreate -c c0t0d0 -n new-zfsBE -p rpool -D /var



If you do not specify `-D /var` with the `lucreate` command, no separate dataset is created for `/var` in the alternate boot environment even if `/var` is a separate file system in the source boot environment.

EXAMPLE 10-1 Migrating From a UFS root (/) File System to ZFS Root Pool

This example shows the same commands as in [Figure 10-1](#). The commands create a new root pool, `rpool`, and create a new boot environment in the pool from a UFS `root (/)` file system. In this example, the `zfs list` command shows the ZFS root pool created by the `zpool` command. The next `zfs list` command shows the datasets created by the `lucreate` command.

EXAMPLE 10-1 Migrating From a UFS root (/) File System to ZFS Root Pool (Continued)

```
# zpool create rpool c0t1d0s5
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                              5.97G  23.3G   31K    /rpool

# lucreate -c c0t0d0 -n new-zfsBE -p rpool
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                              5.97G  23.3G   31K    /rpool
rpool/ROOT                         4.42G  23.3G   31K    legacy
rpool/ROOT/new-zfsBE               4.42G  23.3G  4.42G    /
rpool/dump                         1.03G  24.3G   16K    -
rpool/swap                         530M   23.8G   16K    -
```

The following `zfs list` command shows the separate dataset created for `/var` by using the `-D` `/var` option in the `lucreate` command.

```
# lucreate -c c0t0d0 -n new-zfsBE -p rpool -D /var
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                              5.97G  23.3G   31K    /rpool
rpool/ROOT                         4.42G  23.3G   31K    legacy
rpool/ROOT/new-zfsBE               4.42G  23.3G  4.42G    /
rpool/ROOT/new-zfsBE/var           248MG  23.3G  248M    /var
rpool/dump                         1.03G  24.3G   16K    -
rpool/swap                         530M   23.8G   16K    -
```

The new boot environment is `new-zfsBE`. The boot environment, `new-zfsBE`, is ready to be upgraded and activated.

Migrating a UFS File System With Solaris Volume Manager Volumes Configured to a ZFS Root File System

You can migrate your UFS file system if your system has Solaris Volume Manager (SVM) volumes. To create a UFS boot environment from an existing SVM configuration, you create a new boot environment from your currently running system. Then create the ZFS boot environment from the new UFS boot environment.

EXAMPLE 10-2 Migrating From a UFS root (/) File System With SVM Volumes to ZFS Root Pool

When migrating a system with SVM volumes, the SVM volumes are ignored. You can set up mirrors within the root pool as in the following example.

In this example, the `lucreate` command with the `-m` option creates a new boot environment from the currently running system. The disk slice `c1t0d0s0` contains a UFS root (/) file system configured with SVM volumes. The `zpool` command creates a root pool, `c1t0d0s0`, and a

EXAMPLE 10-2 Migrating From a UFS root (/) File System With SVM Volumes to ZFS Root Pool
(Continued)

RAID-1 volume (mirror), `c0t0d0s0`. In the second `lucreate` command, the `-n` option assigns the name to the boot environment to be created, `c0t0d0s0`. The `-s` option, identifies the UFS root (/) file system. The `-p` option specifies where to place the new boot environment, `rpool`.

```
# lucreate -n ufsBE -m /:/dev/md/dsk/d104:ufs
# zpool create rpool mirror c1t0d0s0 c2t1d0s0
# lucreate -n c0t0d0s0 -s ufsBE -p zpool
```

The boot environment, `c0t0d0s0`, is ready to be upgraded and activated.

Creating a New Boot Environment From a ZFS Root Pool

You can either create a new ZFS boot environment within the same root pool or on a new root pool. This section contains the following overviews:

- [“Creating a New Boot Environment Within the Same Root Pool” on page 172](#)
- [“Creating a New Boot Environment on Another Root Pool” on page 174](#)

Creating a New Boot Environment Within the Same Root Pool

When creating a new boot environment within the same ZFS root pool, the `lucreate` command creates a snapshot from the source boot environment and then a clone is made from the snapshot. The creation of the snapshot and clone is almost instantaneous and the disk space used is minimal. The amount of space ultimately required depends on how many files are replaced as part of the upgrade process. The snapshot is read-only, but the clone is a read-write copy of the snapshot. Any changes made to the clone boot environment are not reflected in either the snapshot or the source boot environment from which the snapshot was made.

Note – As data within the active dataset changes, the snapshot consumes space by continuing to reference the old data. As a result, the snapshot prevents the data from being freed back to the pool. For more information about snapshots, see [Chapter 6, “Working With Oracle Solaris ZFS Snapshots and Clones,”](#) in *Oracle Solaris ZFS Administration Guide*.

When the current boot environment resides on the same ZFS pool, the `-p` option is omitted.

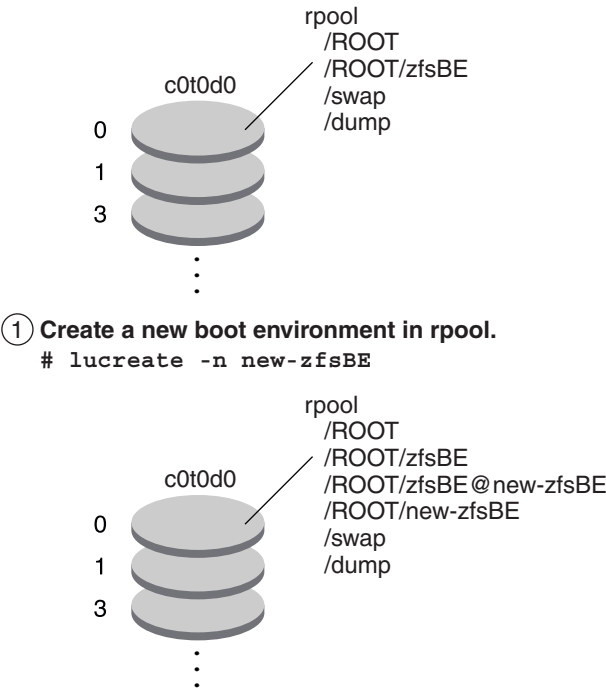
The following figure shows the creation of a ZFS boot environment from a ZFS root pool. The slice `c0t0d0s0` contains a the ZFS root pool, `rpool`. In the `lucreate` command, the `-n` option assigns the name to the boot environment to be created, `new-zfsBE`. A snapshot of the original

root pool is created `rpool@new-zfsBE`. The snapshot is used to make the clone that is a new boot environment, `new-zfsBE`. The boot environment, `new-zfsBE`, is ready to be upgraded and activated.

FIGURE 10-3 Creating a New Boot Environment on the Same Root Pool

Creating a New Boot Environment on the Same Root Pool

Currently Running System



EXAMPLE 10-3 Creating a Boot Environment Within the Same ZFS Root Pool

This example shows the same command as in the figure. It creates a new boot environment in the same root pool. The `lucreate` command names the currently running boot environment with the `-c zfsBE` option, and the `-n new-zfsBE` creates the new boot environment. The `zfs list` command shows the ZFS datasets with the new boot environment and snapshot.

```
# lucreate -c zfsBE -n new-zfsBE
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               11.4G  2.95G   31K    /rpool
rpool/ROOT                          4.34G  2.95G   31K    legacy
rpool/ROOT/new-zfsBE                 4.34G  2.95G  4.34G    /
rpool/dump                           2.06G  5.02G   16K    -
```

EXAMPLE 10-3 Creating a Boot Environment Within the Same ZFS Root Pool (Continued)

rpool/swap	5.04G	7.99G	16K	-
------------	-------	-------	-----	---

Creating a New Boot Environment on Another Root Pool

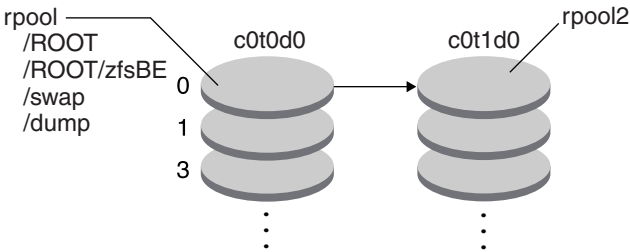
You can use the `lucreate` command to copy an existing ZFS root pool into another ZFS root pool. The copy process might take some time depending on your system.

The following figure shows the `zpool` command that creates a ZFS root pool, `rpool2`, on `c0t1d0s5` because a bootable ZFS root pool does not yet exist. The `lucreate` command with the `-n` option assigns the name to the boot environment to be created, `new-zfsBE`. The `-p` option specifies where to place the new boot environment.

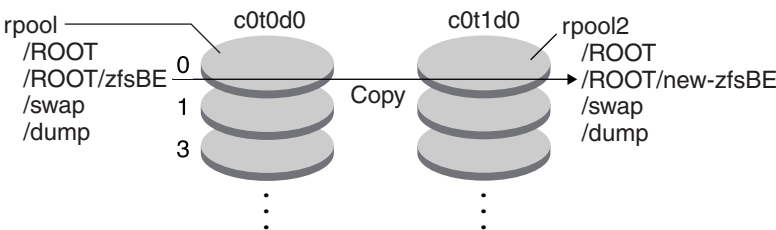
FIGURE 10-4 Creating a New Boot Environment on Another Root Pool

Creating a New Boot Environment in Another Root Pool

- ① Create the new root pool on a separate slice.
`zpool create rpool2 c0t1d0s0`



- ② Create the new boot environment on `rpool2`.
`lucreate -n new-zfsBE -p rpool2`



EXAMPLE 10-4 Creating a Boot Environment on a Different ZFS Root Pool

This example shows the same commands as in the figure which create a new root pool and then a new boot environment in the newly created root pool. In this example, the `zpool create` command creates `rpool2`. The `zfs list` command shows that no ZFS datasets are created in `rpool2`. The datasets are created with the `lucreate` command.

```
# zpool create rpool2 c0t2d0s5
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	11.4G	2.95G	31K	/rpool
rpool/ROOT	4.34G	2.95G	31K	legacy
rpool/ROOT/new-zfsBE	4.34G	2.95G	4.34G	/
rpool/dump	2.06G	5.02G	16K	-
rpool/swap	5.04G	7.99G	16K	-

The new ZFS root pool, `rpool2`, is created on disk slice `c0t2d0s5`.

```
# lucreate -n new-zfsBE -p rpool2
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	11.4G	2.95G	31K	/rpool
rpool/ROOT	4.34G	2.95G	31K	legacy
rpool/ROOT/new-zfsBE	4.34G	2.95G	4.34G	/
rpool/dump	2.06G	5.02G	16K	-
rpool/swap	5.04G	7.99G	16K	-

The new boot environment, `new-zfsBE`, is created on `rpool2` along with the other datasets, `ROOT`, `dump` and `swap`. The boot environment, `new-zfsBE`, is ready to be upgraded and activated.

Creating a New Boot Environment From a Source Other Than the Currently Running System

If you are creating a boot environment from a source other than the currently running system, you must use the `lucreate` command with the `-s` option. The `-s` option works the same as for a UFS file system. The `-s` option provides the path to the alternate root (`/`) file system. This alternate root (`/`) file system is the source for the creation of the new ZFS root pool. The alternate root can be either a UFS (`/`) root file system or a ZFS root pool. The copy process might take time, depending on your system.

EXAMPLE 10-5 Creating a Boot Environment From an Alternate Root (`/`) File System

The following command creates a new ZFS root pool from an existing ZFS root pool. The `-n` option assigns the name to the boot environment to be created, `new-zfsBE`. The `-s` option specifies the boot environment, `source-zfsBE`, to be used as the source of the copy instead of the currently running boot environment. The `-p` option specifies to place the new boot environment in `newpool2`.

```
# lucreate -n new-zfsBE -s source-zfsBE -p rpool2
```

EXAMPLE 10-5 Creating a Boot Environment From an Alternate Root (/) File System (Continued)

The boot environment, new-zfsBE, is ready to be upgraded and activated.

Creating a ZFS Boot Environment on a System With Non-Global Zones Installed

You can use Live Upgrade to migrate your non-global zones to a ZFS root file system. For overview, planning, and step-by-step procedure information, see [Chapter 13, “Live Upgrade for ZFS With Non-Global Zones Installed.”](#)

Live Upgrade for ZFS (Planning)

This chapter provides guidelines and requirements for review before performing a migration of a UFS file system to a ZFS file system or before creating a new ZFS boot environment from an existing ZFS root pool.

Note – Creating boot environments with Live Upgrade, a feature of Oracle Solaris, is new in the Solaris 10 10/08 release. When you perform a Live Upgrade for a UFS file system, both the command-line parameters and operation of Live Upgrade remain unchanged. To perform a Live Upgrade on a system with UFS file systems, see [Part I, “Upgrading With Live Upgrade,”](#) in this book.

System Requirements and Limitations When Using Live Upgrade

Be sure that you have read and understand the following requirements and limitations before performing a migration of a UFS file system to a ZFS file system or before creating a new ZFS boot environment from an existing ZFS root pool. These requirements are in addition to the requirements listed in [Chapter 5, “ZFS Root File System Installation Planning,”](#) in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*.

TABLE 11-1 Requirements and Limitations

Requirement or Limitation	Description	Information
You must have at the least the Solaris 10 10/08 release installed.	Migrating from a UFS file system to a ZFS root pool with Live Upgrade or creating a new boot environment in a root pool is new in the Solaris 10 10/08 release. This release contains the software needed to use Live Upgrade with ZFS. You must have at least this release installed to use ZFS.	

TABLE 11-1 Requirements and Limitations (Continued)

Requirement or Limitation	Description	Information
Disk space	The minimum amount of available pool space for a bootable ZFS root file system depends on the amount of physical memory, the disk space available, and the number of boot environments to be created.	For an explanation, see “Disk Space Requirements for a ZFS Installation” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.
When you migrate from a UFS root (/) file system to a ZFS root pool, consider these requirements.	<ul style="list-style-type: none"> ■ Migration is possible only from a UFS file system to a ZFS file system. <ul style="list-style-type: none"> ■ File systems other than a UFS file system cannot be migrated to a ZFS root pool. ■ A UFS file system cannot be created from a ZFS root pool. ■ Before migrating, a ZFS storage pool must exist. ■ The ZFS storage pool must be created with slices rather than whole disks to be upgradeable and bootable. <ul style="list-style-type: none"> ■ The pool created with slices can be mirrored, but not a RAID-Z or non-redundant configuration of multiple disks. The SVM device information must be already available in the /dev/md/[r]disk directory. ■ The pool must have an SMI label. An EFI-labeled disk cannot be booted. ■ x86 only: The ZFS pool must be in a slice with an fdisk partition. 	<ul style="list-style-type: none"> ■ For step-by-step procedures, see “Migrating a UFS File System to a ZFS File System” on page 181. ■ For information about SVM, see “Overview of Solaris Volume Manager Components” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade. ■ For information about fdisk partitions, see “Partitioning Recommendations” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade. ■ For information about other restrictions for an EFI-labeled disk, see “Overview of Disk Management” in System Administration Guide: Devices and File Systems.
When you migrate shared file systems, shared file systems cannot be copied to a separate slice on the new ZFS root pool.	For example, when performing a Live Upgrade with a UFS root (/) file system, you can use the -m option to copy the /export file system to another device. You do not have the -m option of copying the shared file system to a ZFS pool.	

TABLE 11-1 Requirements and Limitations (Continued)

Requirement or Limitation	Description	Information
When you are migrating a UFS root file system that contains non-global zones, shared file systems are not migrated.	On a system with a UFS root (/) file system and non-global zones installed, the non-global zones are migrated if the zone is in a critical file system as part of the UFS to ZFS migration, or the zone is cloned when you upgrade within the same ZFS pool. If a non-global zone exists in a shared UFS (/) file system, to migrate to a ZFS root pool, you must first upgrade the zone, as in previous Oracle Solaris releases.	<ul style="list-style-type: none"> ■ For an explanation of critical and shared file systems, see “File System Types” on page 22. ■ For step-by-step instructions when non-global zones are installed, see Chapter 13, “Live Upgrade for ZFS With Non-Global Zones Installed.”
Do not use the ZFS rename command.	The Live Upgrade feature is unaware of the name change and subsequent commands, such as <code>ldelete</code> , will fail. Do not rename your ZFS pools or file systems if you have existing boot environments that you want to continue to use.	
Set dataset properties before the <code>lucreate</code> command is used.	Live Upgrade creates the datasets for the boot environment and ZFS volumes for the swap area and dump device but does not account for any existing dataset property modifications. Therefore if you want a dataset property enabled in the new boot environment, you must set the property before the <code>lucreate</code> operation. For example: # zfs set compression=on rpool/ROOT	See “Introducing ZFS Properties” in <i>Oracle Solaris ZFS Administration Guide</i> .
When creating a ZFS boot environment within the same ZFS root pool, you cannot use the <code>lucreate</code> command include and exclude options to customize the content.	You cannot use the <code>-f</code> , <code>-o</code> , <code>-y</code> , <code>-Y</code> , and <code>-z</code> options to include or exclude files from the primary boot environment when creating a boot environment in the same ZFS root pool. However, you can use these options in the following cases: <ul style="list-style-type: none"> ■ Creating a boot environment from a UFS file system to a UFS file system ■ Creating a boot environment from a UFS file system to a ZFS root pool ■ Creating a boot environment from a ZFS root pool to a different ZFS root pool 	For information about using the include and exclude options, see “Creating a Boot Environment and Customizing the Content” on page 83.

Note – You cannot use Live Upgrade to upgrade non-root ZFS file systems.

Creating a Boot Environment for ZFS Root Pools

This chapter provides step-by-step procedures on how to create a ZFS boot environment when you use Live Upgrade.

Note – Migrating from a UFS file system to a ZFS root pool or creating ZFS boot environments with Live Upgrade is new in the Solaris 10 10/08 release. To find out how to use Live Upgrade on a system with UFS file systems, see [Part I, “Upgrading With Live Upgrade,”](#) of this book.

This chapter provides procedures for the following tasks:

- “Migrating a UFS File System to a ZFS File System” on page 181
- “Creating a Boot Environment Within the Same ZFS Root Pool” on page 188
- “Creating a Boot Environment In a New Root Pool” on page 192
- “Creating a Boot Environment From a Source Other Than the Currently Running System” on page 197
- “Falling Back to a ZFS Boot Environment” on page 198

For procedures on using ZFS when non-global zones are installed, see [Chapter 13, “Live Upgrade for ZFS With Non-Global Zones Installed.”](#)

Migrating a UFS File System to a ZFS File System

This procedure describes how to migrate a UFS file system to a ZFS file system. Creating a boot environment provides a method of copying critical file systems from an active UFS boot environment to a ZFS root pool. The `lucreate` command copies the critical file systems to a new boot environment within an existing ZFS root pool. User-defined (shareable) file systems are not copied and are not shared with the source UFS boot environment. Also, `/swap` is not shared between the UFS file system and ZFS root pool. For an overview of critical and shareable file systems, see [“File System Types”](#) on page 22.

▼ How to Migrate a UFS File System to a ZFS File System

Note – To migrate an active UFS root (/) file system to a ZFS root pool, you must provide the name of the root pool. The critical file systems are copied into the root pool.

- 1 Before running Live Upgrade for the first time, you must install the latest Live Upgrade packages from installation media and install the patches listed in the My Oracle Support knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on the [My Oracle Support](#) web site.**

The latest packages and patches ensure that you have all the latest bug fixes and new features in the release. Ensure that you install all the patches that are relevant to your system before proceeding to create a new boot environment.

The following substeps describe the steps in the My Oracle Support knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

Note – Using Live Upgrade to create new ZFS boot environments requires at least the Solaris 10 10/08 release to be installed. Previous releases do not have the ZFS and Live Upgrade software to perform the tasks.

- a. Become superuser or assume an equivalent role.**

Note – Roles contain authorizations and privileged commands. For more information about roles, see “[Configuring RBAC \(Task Map\)](#)” in *System Administration Guide: Security Services*.

- b. From the My Oracle Support web site, follow the instructions in knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) to remove and add Live Upgrade packages.**

The three Live Upgrade packages, SUNWluu, SUNWlur, and SUNWlucfg, comprise the software needed to upgrade by using Live Upgrade. These packages include existing software, new features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading to the target release fails. The SUNWlucfg package is new **starting with the Solaris 10 8/07 release**. If you are using Live Upgrade packages from a release previous to Solaris 10 8/07, you do not need to remove this package.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

- c. Install the new Live Upgrade packages from the release to which you are upgrading. For instructions, see “[Installing Live Upgrade](#)” on page 60.**

d. **Before running Live Upgrade, you are required to install the following patches. These patches ensure that you have all the latest bug fixes and new features in the release.**

Ensure that you have the most recently updated patch list by consulting [My Oracle Support](#). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on the My Oracle Support web site.

- If you are storing the patches on a local disk, create a directory such as `/var/tmp/lupatches` and download the patches to that directory.
- From the [My Oracle Support](#) web site, obtain the list of patches.
- Change to the patch directory.

```
# cd /var/tmp/lupatches
```

- Install the patches with the `patchadd` command.

```
# patchadd patch_id
```

patch_id is the patch number or numbers. Separate multiple patch names with a space.

Note – The patches need to be applied in the order that is specified in knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

- Reboot the system if necessary. Certain patches require a reboot to be effective.
x86 only: Rebooting the system is required or Live Upgrade fails.

```
# init 6
```

You now have the packages and patches necessary for a successful migration.

2 Create a ZFS root pool.

The ZFS root pool must be on a single slice to be bootable and upgradeable.

```
# zpool create rpool c0t1d0s5
```

rpool Specifies the name of the new ZFS root pool to be created.

c0t1d0s5 Creates the new root pool on the disk slice, *c0t1d0s5*.

For information about creating a new root pool, see the [Oracle Solaris ZFS Administration Guide](#).

3 Migrate your UFS root (/) file system to the new ZFS root pool.

```
# lucreate [-c ufsBE] -n new-zfsBE -p rpool
```

ufsBE The name for the current UFS boot environment. This option is not required and is used only when the first boot environment is created. If you run the `lucreate` command for the first time and you omit the `-c` option, the software creates a default name for you.

- new-zfsBE*

The name for the boot environment to be created. The name must be unique on the system.
- p rpool*

Places the newly created ZFS root (/) file system into the ZFS root pool defined in *rpool*.

The creation of the new ZFS boot environment might take a while. The UFS file system data is being copied to the ZFS root pool. When the inactive boot environment has been created, you can use the `luupgrade` or `luactivate` command to upgrade or activate the new ZFS boot environment.

4 (Optional) Verify that the boot environment is complete.

```
# lustatus
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
ufsBE	yes	yes	yes	no	-
new-zfsBE	yes	no	no	yes	-

5 (Optional) Verify the basic dataset information on the system.

The `list` command displays the names of all datasets on the system. In this example, `rpool` is the name of the ZFS pool and `new-zfsBE` is the name of the newly created ZFS boot environment.

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	11.4G	2.95G	31K	/rpool
rpool/ROOT	4.34G	2.95G	31K	legacy
rpool/ROOT/new-zfsBE	4.34G	2.95G	4.34G	/
rpool/dump	2.06G	5.02G	16K	-
rpool/swap	5.04G	7.99G	16K	-

The mount points listed for the new boot environment are temporary until the `luactivate` command is executed. The `/dump` and `/swap` volumes are not shared with the original UFS boot environment, but are shared within the ZFS root pool and boot environments within the root pool.

You can now upgrade and activate the new boot environment.

Example 12-1 Migrating a UFS Root (/) File System to a ZFS Root Pool

In this example, the new ZFS root pool, `rpool`, is created on a separate slice, `C0t0d0s4`. The `lucreate` command migrates the currently running UFS boot environment, `c0t0d0`, to the new ZFS boot environment, `new-zfsBE`, and places the new boot environment in `rpool`.

```
# zpool create rpool C0t0d0s4
```

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	9.29G	57.6G	20K	/rpool


```
# lucreate -c c0t0d0 -n new-zfsBE -p rpool
Determining types of file systems supported
Validating file system requests
Preparing logical storage devices
Preparing physical storage devices
Configuring physical storage devices
Configuring logical storage devices
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <c0t0d0>.
Creating initial configuration for primary boot environment <c0t0d0>.
INFORMATION: No BEs are configured on this system.
The device </dev/dsk/clt0d0s0> is not a root device for any boot
environment; cannot get BE ID.
PBE configuration successful: PBE name <c0t0d0> PBE Boot Device
</dev/dsk/clt0d0s0>.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/clt0d0s6> is not a root device for any boot
environment; cannot get BE ID.
Creating configuration for boot environment <new-zfsBE>.
Source boot environment is <c0t0d0>.
Creating file systems on boot environment <new-zfsBE>.
Creating <zfs> file system for </> in zone <global> on
<rpool/ROOT/new-zfsBE>.
Populating file systems on boot environment <new-zfsBE>.
Analyzing zones.
Mounting ABE <new-zfsBE>.
Generating file list.
Copying data from PBE <c0t0d0> to ABE <new-zfsBE>.
100% of filenames transferred
Finalizing ABE.
Fixing zonepaths in ABE.
Unmounting ABE <new-zfsBE>.
Fixing properties on ZFS datasets in ABE.
Reverting state of zones in PBE <c0t0d0>.
Making boot environment <new-zfsBE> bootable.
Creating boot_archive for /.alt.tmp.b-Cjh.mnt
updating /.alt.tmp.b-Cjh.mnt/platform/sun4u/boot_archive
Population of boot environment <new-zfsBE> successful.
Creation of boot environment <new-zfsBE> successful.
```

lustatus

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
c0t0d0	yes	yes	yes	no	-
new-zfsBE	yes	no	no	yes	-

zfs list

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	11.4G	2.95G	31K	/rpool
rpool/ROOT	4.34G	2.95G	31K	legacy
rpool/ROOT/new-zfsBE	4.34G	2.95G	4.34G	/
rpool/dump	2.06G	5.02G	16K	-
rpool/swap	5.04G	7.99G	16K	-

You can now upgrade or activate the new boot environment.

In this example, the new boot environment is upgraded by using the `luupgrade` command from an image that is stored in the location indicated with the `-s` option.

```
# luupgrade -n zfsBE -u -s /net/install/export/s10/combined.s10
 51135 blocks
miniroot filesystem is <lofs>
Mounting miniroot at
</net/install/export/solaris_10/combined.solaris_10_wos
/Solaris_10/Tools/Boot>
Validating the contents of the media
</net/install/export/s10/combined.s10>.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains Solaris version <10_1008>.
Constructing upgrade profile to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live
Upgrade requests.
Creating upgrade profile for BE <zfsBE>.
Determining packages to install or upgrade for BE <zfsBE>.
Performing the operating system upgrade of the BE <zfsBE>.
CAUTION: Interrupting this process may leave the boot environment
unstable or unbootable.
Upgrading Solaris: 100% completed
Installation of the packages from this media is complete.
Adding operating system patches to the BE <zfsBE>.
The operating system patch installation is complete.
INFORMATION: The file /var/sadm/system/logs/upgrade_log on boot
environment <zfsBE> contains a log of the upgrade operation.
INFORMATION: The file var/sadm/system/data/upgrade_cleanup on boot
environment <zfsBE> contains a log of cleanup operations required.
INFORMATION: Review the files listed above. Remember that all
of the files are located on boot environment <zfsBE>.
Before you activate boot environment <zfsBE>, determine if any
additional system maintenance is required or if additional media
of the software distribution must be installed.
The Solaris upgrade of the boot environment <zfsBE> is complete.
```

The new boot environment can be activated anytime after it is created.

```
# luactivate new-zfsBE
```

A Live Upgrade Sync operation will be performed on startup of boot environment <new-zfsBE>.

```
*****
```

The target boot environment has been activated. It will be used when you reboot. NOTE: You MUST NOT USE the `reboot`, `halt`, or `uadmin` commands. You MUST USE either the `init` or the `shutdown` command when you reboot. If you do not use either `init` or `shutdown`, the system will not boot using the target BE.

```
*****
```

In case of a failure while booting to the target BE, the following process needs to be followed to fallback to the currently working boot environment:

1. Enter the PROM monitor (ok prompt).
2. Boot the machine to Single User mode using a different boot device (like the Solaris Install CD or Network). Examples:

```
At the PROM monitor (ok prompt):
For boot to Solaris CD:  boot cdrom -s
For boot to network:    boot net -s
```

3. Mount the Current boot environment root slice to some directory (like /mnt). You can use the following command to mount:

```
mount -Fufs /dev/dsk/clt0d0s0 /mnt
```

4. Run <luactivate> utility with out any arguments from the current boot environment root slice, as shown below:

```
/mnt/sbin/luactivate
```

5. luactivate, activates the previous working boot environment and indicates the result.

6. Exit Single User mode and reboot the machine.

```
*****
```

```
Modifying boot archive service
Activation of boot environment <new-zfsBE> successful.
```

Reboot the system to the ZFS boot environment.

```
# init 6
# svc.startd: The system is coming down. Please wait.
svc.startd: 79 system services are now being stopped.
.
.
.
```

If you fall back to the UFS boot environment, then you need to import again any ZFS storage pools that were created in the ZFS boot environment because they are not automatically available in the UFS boot environment. You will see messages similar to the following example when you switch back to the UFS boot environment.

```
# luactivate c0t0d0
WARNING: The following files have changed on both the current boot
environment <new-zfsBE> zone <global> and the boot environment
to be activated <c0t0d0>:
/etc/zfs/zpool.cache
INFORMATION: The files listed above are in conflict between the current
boot environment <zfsBE> zone <global> and the boot environment to be
activated <c0t0d0>. These files will not be automatically synchronized
```

from the current boot environment <new-zfsBE> when boot environment <c0t0d0>

Creating a Boot Environment Within the Same ZFS Root Pool

The following procedure describes how to create a new ZFS boot environment within an existing ZFS root pool. After the inactive boot environment is created, the new boot environment can be upgraded and activated at your convenience. The `-p` option is not required when you create a boot environment within the same pool.

▼ How to Create a ZFS Boot Environment Within the Same ZFS Root Pool

- 1 Before running Live Upgrade for the first time, you must install the latest Live Upgrade packages from installation media and install the patches listed in the My Oracle Support knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on the [My Oracle Support](#) web site.

The latest packages and patches ensure that you have all the latest bug fixes and new features in the release. Ensure that you install all the patches that are relevant to your system before proceeding to create a new boot environment.

The following substeps describe the steps in the My Oracle Support knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

Note – Using Live Upgrade to create new ZFS boot environments requires at least the Solaris 10 10/08 release to be installed. Previous releases do not have the ZFS and Live Upgrade software to perform the tasks.

a. Become superuser or assume an equivalent role.

Note – Roles contain authorizations and privileged commands. For more information about roles, see “[Configuring RBAC \(Task Map\)](#)” in *System Administration Guide: Security Services*.

- b. From the My Oracle Support web site, follow the instructions in knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) to remove and add Live Upgrade packages.

The three Live Upgrade packages, `SUNWluu`, `SUNWlur`, and `SUNWlucfg`, comprise the software needed to upgrade by using Live Upgrade. These packages include existing software, new

features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading to the target release fails. The SUMWLucfg package is new **starting with the Solaris 10 8/07 release**. If you are using Live Upgrade packages from a release previous to Solaris 10 8/07, you do not need to remove this package.

Note – The SUMWLucfg package is new **starting with the Solaris 10 8/07 release**. If you are using Live Upgrade packages from a previous release, you do not need to remove this package.

```
# pkgrm SUNWLucfg SUNWluu SUNWlur
```

- c. **Install the new Live Upgrade packages. For instructions, see “Installing Live Upgrade” on page 60.**
- d. **Before running Live Upgrade, you are required to install the following patches. These patches ensure that you have all the latest bug fixes and new features in the release.**
Ensure that you have the most recently updated patch list by consulting [My Oracle Support](#). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on the My Oracle Support web site.
 - If you are storing the patches on a local disk, create a directory such as `/var/tmp/lupatches` and download the patches to that directory.
 - From the [My Oracle Support](#) web site, obtain the list of patches.
 - Change to the patch directory as in this example.


```
# cd /var/tmp/lupatches
```
 - Install the patches with the `patchadd` command.


```
# patchadd -M path-to-patches patch_id patch_id
```

path-to-patches is the path to the patch directory such as `/var/tmp/lupatches`.
patch_id is the patch number or numbers. Separate multiple patch names with a space.

Note – The patches need to be applied in the order that is specified in the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844).

- Reboot the system if necessary. Certain patches require a reboot to be effective.
x86 only: Rebooting the system is required or Live Upgrade fails.


```
# init 6
```

You now have the packages and patches necessary for a successful creation of a new boot environment.

2 Create the new boot environment.

```
# lucreate [-c zfsBE] -n new-zfsBE
```

zfsBE The name for the current boot environment. This option is not required and is used only when the first boot environment is created. If you run `lucreate` for the first time and you omit the `-c` option, the software creates a default name for you.

new-zfsBE The name for the boot environment to be created. The name must be unique on the system.

The creation of the new boot environment is almost instantaneous. A snapshot is created of each dataset in the current ZFS root pool, and a clone is then created from each snapshot. Snapshots are very disk-space efficient, and this process uses minimal disk space. When the inactive boot environment has been created, you can use the `luupgrade` or `luactivate` command to upgrade or activate the new ZFS boot environment.

3 (Optional) Verify that the boot environment is complete.

The `lustatus` command reports whether the boot environment creation is complete and bootable.

```
# lustatus
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
zfsBE	yes	yes	yes	no	-
new-zfsBE	yes	no	no	yes	-

4 (Optional) Verify the basic dataset information on the system.

In this example, the ZFS root pool is named `rpool`, and the `@` symbol indicates a snapshot. The new boot environment mount points are temporary until the `luactivate` command is executed. The `/dump` and `/swap` volumes are shared with the ZFS root pool and boot environments within the root pool.

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	11.4G	2.95G	31K	/rpool
rpool/ROOT	4.34G	2.95G	31K	legacy
rpool/ROOT/new-zfsBE	4.34G	2.95G	4.34G	/
rpool/dump	2.06G	5.02G	16K	-
rpool/swap	5.04G	7.99G	16K	-

You can now upgrade and activate the new boot environment.

Example 12-2 Creating a Boot Environment Within the Same ZFS Root Pool

This example creates a new ZFS boot environment, `new-zfsBE`. The `-p` option is not required because the boot environment is being created within the same root pool.

```
# lucreate [-c zfsBE] -n new-zfsBE
Analyzing system configuration.
Comparing source boot environment <zfsBE> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Creating configuration for boot environment new-zfsBE.
Source boot environment is zfsBE.
Creating boot environment new-zfsBE.
Cloning file systems from boot environment zfsBE to create
boot environment new-zfsBE.
Creating snapshot for <rpool> on <rpool> Creating clone for <rpool>.
Setting canmount=noauto for <rpool> in zone <global> on <rpool>.
Population of boot environment zfsBE successful on <rpool>.
# lustatus
boot environment      Is      Active      Active      Can      Copy
Name                 Complete Now      OnReboot    Delete    Status
-----
zfsBE                 yes     yes       yes         no        -
new-zfsBE             yes     no        no          yes       -
# zfs list
NAME                                     USED  AVAIL  REFER  MOUNTPOINT
rpool                                11.4G  2.95G   31K    /rpool
rpool/ROOT                      4.34G  2.95G   31K    legacy
rpool/ROOT/new-zfsBE            4.34G  2.95G  4.34G    /
rpool/dump                      2.06G  5.02G   16K    -
rpool/swap                     5.04G  7.99G   16K    -
```

You can now upgrade and activate the new boot environment. For an example of upgrading a ZFS boot environment, see [Example 12-1](#). For more examples of using the `luupgrade` command, see [Chapter 5, “Upgrading With Live Upgrade \(Tasks\)”](#).

```
# luactivate new-zfsBE
```

A Live Upgrade Sync operation will be performed on startup of boot environment `<new-zfsBE>`.

```
*****
```

The target boot environment has been activated. It will be used when you reboot. NOTE: You MUST NOT USE the `reboot`, `halt`, or `uadmin` commands. You MUST USE either the `init` or the `shutdown` command when you reboot. If you do not use either `init` or `shutdown`, the system will not boot using the target BE.

```
*****
```

In case of a failure while booting to the target BE, the following process needs to be followed to fallback to the currently working boot environment:

1. Enter the PROM monitor (ok prompt).
2. Boot the machine to Single User mode using a different boot device (like the Solaris Install CD or Network). Examples:

 At the PROM monitor (ok prompt):
 For boot to Solaris CD: boot cdrom -s
 For boot to network: boot net -s
3. Mount the Current boot environment root slice to some directory (like /mnt). You can use the following command to mount:


```
mount -Fufs /dev/dsk/clt0d0s0 /mnt
```
4. Run <luactivate> utility with out any arguments from the current boot environment root slice, as shown below:


```
/mnt/sbin/luactivate
```
5. luactivate, activates the previous working boot environment and indicates the result.
6. Exit Single User mode and reboot the machine.

Modifying boot archive service
 Activation of boot environment <new-zfsBE> successful.

Reboot the system to the ZFS boot environment.

```
# init 6
# svc.startd: The system is coming down. Please wait.
svc.startd: 79 system services are now being stopped.
.
.
.
```

Creating a Boot Environment In a New Root Pool

If you have an existing ZFS root pool and want to create a new ZFS boot environment in a new root pool, the following procedure provides the steps. After the inactive boot environment is created, the new boot environment can be upgraded and activated at your convenience. The -p option is required to note where to place the new boot environment. The existing ZFS root pool must exist and be on a separate slice to be bootable and upgradeable.

▼ How to Create a Boot Environment on a New ZFS Root Pool

- 1 Before running Live Upgrade for the first time, you must install the latest Live Upgrade packages from installation media and install the patches listed in the knowledge document. Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on the [My Oracle Support](#) web site.

The latest packages and patches ensure that you have all the latest bug fixes and new features in the release. Ensure that you install all the patches that are relevant to your system before proceeding to create a new boot environment.

The following substeps describe the steps in the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

Note – Using Live Upgrade to create new ZFS boot environments requires at least the Solaris 10 10/08 release to be installed. Previous releases do not have the ZFS and Live Upgrade software to perform the tasks.

a. Become superuser or assume an equivalent role.

Note – Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in *System Administration Guide: Security Services*.

- b. From the My Oracle Support web site, follow the instructions in the knowledge document 1004881.1 - to remove and add Live Upgrade packages.

The three Live Upgrade packages, SUNWluu, SUNWlur, and SUNWlucfg, comprise the software needed to upgrade by using Live Upgrade. These packages include existing software, new features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading to the target release fails. The SUNWlucfg package is new **starting with the Solaris 10 8/07 release**. If you are using Live Upgrade packages from a release previous to Solaris 10 8/07, you do not need to remove this package.

Note – The SUNWlucfg package is new **starting with the Solaris 10 8/07 release**. If you are using Live Upgrade packages from a previous release, you do not need to remove this package.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

c. Install the new Live Upgrade packages. For instructions, see [“Installing Live Upgrade” on page 60](#).

d. Before running Live Upgrade, you are required to install the following patches. These patches ensure that you have all the latest bug fixes and new features in the release.

Ensure that you have the most recently updated patch list by consulting [My Oracle Support](#). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

- If you are storing the patches on a local disk, create a directory such as `/var/tmp/lupatches` and download the patches to that directory.
- From the [My Oracle Support](#) web site, obtain the list of patches.
- Change to the patch directory as in this example.

```
# cd /var/tmp/lupatches
```

- Install the patches with the `patchadd` command.

```
# patchadd -M path-to-patches patch_id patch_id
```

path-to-patches is the path to the patch directory such as `/var/tmp/lupatches`.

patch_id is the patch number or numbers. Separate multiple patch names with a space.

Note – The patches need to be applied in the order that is specified in the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

- Reboot the system if necessary. Certain patches require a reboot to be effective.
x86 only: Rebooting the system is required or Live Upgrade fails.

```
# init 6
```

You now have the packages and patches necessary for a successful migration.

2 Create a ZFS root pool.

The ZFS root pool must be on a single slice to be bootable and upgradeable.

```
# zpool create rpool2 c0t1d0s5
```

rpool2 Names of the new ZFS root pool.

c0t1d0s5 Specifies to place *rpool2* on the bootable slice, *c0t1d0s5*.

For information about creating a new root pool, see the [Oracle Solaris ZFS Administration Guide](#).

3 Create the new boot environment.

```
# lucreate [-c zfsBE] -n new-zfsBE -p rpool2
```

zfsBE The name for the current ZFS boot environment.

new-zfsBE The name for the boot environment to be created. The name must be unique on the system.

-p rpool2 Places the newly created ZFS root boot environment into the ZFS root pool defined in *rpool2*.

The creation of the new ZFS boot environment might take a while. The file system data is being copied to the new ZFS root pool. When the inactive boot environment has been created, you can use the `luupgrade` or `luactivate` command to upgrade or activate the new ZFS boot environment.

4 (Optional) Verify that the boot environment is complete.

The `lustatus` command reports whether the boot environment creation is complete and bootable.

```
# lustatus
boot environment   Is      Active  Active  Can      Copy
Name              Complete Now    OnReboot Delete   Status
-----
zfsBE              yes     yes     yes     yes     no      -
new-zfsBE          yes     no      no      no      yes     -
```

5 (Optional) Verify the basic dataset information on the system.

The following example displays the names of all datasets on the system. The mount point listed for the new boot environment are temporary until the `luactivate` command is executed. The new boot environment shares the volumes, `rpool2/dump` and `rpool2/swap`, with the `rpool2` ZFS boot environment.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               11.4G  2.95G   31K    /rpool
rpool/ROOT                          4.34G  2.95G   31K    legacy
rpool/ROOT/new-zfsBE               4.34G  2.95G  4.34G    /
rpool/dump                         2.06G  5.02G   16K    -
rpool/swap                         5.04G  7.99G   16K    -
```

You can now upgrade and activate the new boot environment.

Example 12-3 Creating a Boot Environment on a New Root Pool

In this example, a new ZFS root pool, `newPool`, is created on a separate slice, `c0t1s0s5`. The `lucreate` command creates a new ZFS boot environment, `new-zfsbe`. The `-p` option is required because the boot environment is being created in a different root pool.

```
# zpool create newPool C0t2d0s5
```

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
newPool	92.5K	18.7G	31K	/newPool
rpool		11.4G	2.95G	31K /rpool
rpool/ROOT		4.34G	2.95G	31K legacy
rpool/ROOT/zfsBE	4.34G	2.95G	4.34G	/
rpool/dump		2.06G	5.02G	16K -
rpool/swap		5.04G	7.99G	16K -

```
# lucreate -c c0t1d0s5 -n new-zfsbe -p newPool
```

```
Checking GRUB menu...
```

```
Analyzing system configuration.
```

```
Updating boot environment description database on all BEs.
```

```
Updating system configuration files.
```

```
The device </dev/dsk/ct1d0s0> is not a root device for any boot environment; cannot get BE ID.
```

```
Creating configuration for boot environment <new-zfsbe>.
```

```
Source boot environment is <c0t1d0s5>.
```

```
Creating file systems on boot environment <new-zfsbe>.
```

```
Creating <zfs> file system for </> in zone <global> on <newPool/ROOT/new-zfsbe>.
```

```
Populating file systems on boot environment <new-zfsbe>.
```

```
Analyzing zones.
```

```
Mounting ABE <new-zfsbe>.
```

```
Generating file list.
```

```
Copying data from PBE <c0t1d0s5> to ABE <new-zfsbe>.
```

```
100% of filenames transferred
```

```
Finalizing ABE.
```

```
Fixing zonepaths in ABE.
```

```
Unmounting ABE <new-zfsbe>.
```

```
Fixing properties on ZFS datasets in ABE.
```

```
Reverting state of zones in PBE <c0t1d0s5>.
```

```
Making boot environment <new-zfsbe> bootable.
```

```
Updating bootenv.rc on ABE <new-zfsbe>.
```

```
Saving existing file </boot/grub/menu.lst> in top level dataset for BE <new-zfsbe> as <mount-point>//boot/grub/menu.lst
```

```
File </boot/grub/menu.lst> propagation successful
```

```
Copied GRUB menu from PBE to ABE
```

```
No entry for BE <new-zfsbe> in GRUB menu
```

```
Population of boot environment <new-zfsbe> successful.
```

```
Creation of boot environment <new-zfsbe> successful.
```

```
# lustatus
```

Boot Environment Name	Is Complete	Active Now	Active On Reboot	Can Delete	Copy Status
c0t0d0	yes	yes	yes	no	-
zfsBE		yes	no	no	yes -
new-zfsbe	yes	no	no	yes	-

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
newPool	7.15G	11.6G	36K	/newPool
newPool/ROOT	4.05G	11.6G	31K	legacy
newPool/ROOT/new-zfsbe	4.05G	11.6G	4.05G	/
newPool/dump	1.03G	12.6G	16K	-
newPool/swap	2.06G	13.6G	16K	-
rpool		11.4G	2.95G	31K /rpool
rpool/ROOT		4.34G	2.95G	31K legacy
rpool/ROOT/zfsBE		4.34G	2.95G	4.34G /
rpool/dump		2.06G	5.02G	16K -
rpool/swap		5.04G	7.99G	16K -

Creating a Boot Environment From a Source Other Than the Currently Running System

If you have an existing ZFS root pool or UFS boot environment that is not currently used as the active boot environment, you can use the following example to create a new ZFS boot environment from this boot environment. After the new ZFS boot environment is created, this new boot environment can be upgraded and activated at your convenience.

If you are creating a boot environment from a source other than the currently running system, you must use the `lucreate` command with the `-s` option. The `-s` option works the same as for a UFS file system. The `-s` option provides the path to the alternate root (`/`) file system. This alternate root (`/`) file system is the source for the creation of the new ZFS root pool. The alternate root can be either a UFS (`/`) root file system or a ZFS root pool. The copy process might take time, depending on your system.

The following example shows how the `-s` option is used when creating a boot environment on another ZFS root pool.

EXAMPLE 12-4 How to Create a Boot Environment From a Source Other Than the Currently Running System

The following command creates a new ZFS root pool from an existing ZFS root pool. The `-n` option assigns the name to the boot environment to be created, `new-zfsBE`. The `-s` option specifies the boot environment, `rpool3`, to be used as the source of the copy instead of the currently running boot environment. The `-p` option specifies to place the new boot environment in `rpool2`.

```
# lucreate -n new-zfsBE -s rpool3 -p rpool2
# lustatus
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
zfsBE	yes	yes	yes	no	-
zfsBE2	yes	no	no	yes	-
zfsBE3	yes	no	no	yes	-
new-zfsBE	yes	no	no	yes	-

```
# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	11.4G	2.95G	31K	/rpool
rpool/ROOT	4.34G	2.95G	31K	legacy
rpool/ROOT/new-zfsBE	4.34G	2.95G	4.34G	/
rpool/dump	2.06G	5.02G	16K	-
rpool/swap	5.04G	7.99G	16K	-

You can now upgrade and activate the new boot environment.

Falling Back to a ZFS Boot Environment

If a failure is detected after upgrading or if the application is not compatible with an upgraded component, you can fall back to the original boot environment with the `luactivate` command.

When you have migrated to a ZFS root pool from a UFS boot environment and you then decide to fall back to the UFS boot environment, you again need to import any ZFS storage pools that were created in the ZFS boot environment. These ZFS storage pools are not automatically available in the UFS boot environment. You will see messages similar to the following example when you switch back to the UFS boot environment.

```
# luactivate c0t0d0
```

```
WARNING: The following files have changed on both the current boot
environment <new-ZFSbe> zone <global> and the boot environment
to be activated <c0t0d0>: /etc/zfs/zpool.cache
INFORMATION: The files listed above are in conflict between the current
boot environment <ZFSbe> zone <global> and the boot environment to be
activated <c0t0d0>. These files will not be automatically synchronized
from the current boot environment <new-ZFSbe> when boot
environment <c0t0d0>
```

For examples of falling back to the original boot environment, see [Chapter 6, “Failure Recovery: Falling Back to the Original Boot Environment \(Tasks\)”](#).

Live Upgrade for ZFS With Non-Global Zones Installed

This chapter provides an overview and step-by-step procedures for migrating a UFS (/) root file system to a ZFS root pool. It contains the following information:

- “Creating a ZFS Boot Environment on a System With Non-Global Zones Installed (Overview and Planning)” on page 199
- “Migrating From a UFS Root (/) File System With Non-Global Zones Installed to ZFS Root Pool (Tasks)” on page 200

Note – Migrating from a UFS root (/) file system to a ZFS root pool or creating ZFS boot environments with Live Upgrade, a feature of Oracle Solaris, is new in the Solaris 10 10/08 release. When you perform a Live Upgrade for a UFS file system, both the command-line parameters and operation of Live Upgrade remain unchanged. To perform a Live Upgrade on a system with UFS file systems, see [Part I, “Upgrading With Live Upgrade,”](#) of this book.

Creating a ZFS Boot Environment on a System With Non-Global Zones Installed (Overview and Planning)

You can use Live Upgrade to migrate your UFS root (/) file system with non-global zones installed on a ZFS root pool. All non-global zones that are associated with the file system are also copied to the new boot environment. The non-global zone migration scenarios listed in the following table are supported.

Pre-Migration Root File System and Zone Combination	Post-Migration Root File System and Zone Combination
UFS root file system with the non-global zone root directory in the UFS file system	UFS root file system with the non-global zone root directory in the UFS root file system
	ZFS root pool with the non-global zone root directory in the ZFS root pool

Pre-Migration Root File System and Zone Combination	Post-Migration Root File System and Zone Combination
	ZFS root pool with the non-global zone root directory in a UFS file system
UFS root file system with a non-global zone root in a ZFS root pool	ZFS root pool with the non-global zone root in a ZFS root pool
	UFS root file system with the non-global zone root in ZFS root pool
ZFS root pool with a non-global zone root directory in a ZFS root pool	ZFS root pool with the non-global zone root directory in the ZFS root pool

On a system with a UFS root (/) file system and non-global zones installed, the non-global zones are migrated if the zone is in a non-shared file system as part of the UFS to ZFS migration, or the zone is cloned when you upgrade within the same ZFS pool. If a non-global zone exists in a shared UFS file system, to migrate to another ZFS root pool, you must first upgrade the non-global zone, as in previous Solaris releases.

- For more planning information when migrating to a ZFS root pool, see [“System Requirements and Limitations When Using Live Upgrade” on page 177](#).
- For more limitations on ZFS and non-global zones, see [“Using ZFS on a Solaris System With Zones Installed” in Oracle Solaris ZFS Administration Guide](#).

Migrating From a UFS Root (/) File System With Non-Global Zones Installed to ZFS Root Pool (Tasks)

This section provides step-by-step instructions for migrating from a UFS root (/) file system to a ZFS root pool on a system with non-global zones installed. No non-global zones are on a shared file system in the UFS file system.

▼ How to Migrate a UFS File System to a ZFS Root Pool on a System With Non-Global Zones

The `lucreate` command creates a boot environment of a ZFS root pool from a UFS root (/) file system. A ZFS root pool must exist before the `lucreate` operation and must be created with slices rather than whole disks to be upgradeable and bootable. This procedure shows how an existing non-global zone associated with the UFS root (/) file system is copied to the new boot environment in a ZFS root pool.

- 1 Complete the following steps the first time you perform a Live Upgrade.

Note – Using Live Upgrade to create new ZFS boot environments requires at least the **Solaris 10 10/08 release** to be installed. Previous releases do not have the ZFS and Live Upgrade software to perform the tasks.

- a. **Remove existing Live Upgrade packages on your system if necessary. If you are upgrading to a new release, you must install the packages from that release.**

The three Live Upgrade packages, SUNWluu, SUNWlur, and SUNWlucfg, comprise the software needed to upgrade by using Live Upgrade. These packages include existing software, new features, and bug fixes. If you do not remove the existing packages and install the new packages on your system before using Live Upgrade, upgrading to the target release fails.

```
# pkgrm SUNWlucfg SUNWluu SUNWlur
```

- b. **Install the new Live Upgrade packages from the release to which you are upgrading. For instructions, see “Installing Live Upgrade” on page 60.**

- c. **Before installing or running Live Upgrade, you are required to install the following patches. These patches ensure that you have all the latest bug fixes and new features in the release.**

Ensure that you have the most recently updated patch list by consulting [My Oracle Support](#). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

- Become superuser or assume an equivalent role.

Note – Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in *System Administration Guide: Security Services*.

- If you are storing the patches on a local disk, create a directory such as `/var/tmp/lupatches` and download the patches to that directory.
- From the [My Oracle Support](#) web site, obtain the list of patches.
- Change to the patch directory.

```
# cd /var/tmp/lupatches
```

- Install the patches with the `patchadd` command.

```
# patchadd patch_id
```

patch_id is the patch number or numbers. Separate multiple patch names with a space.

Note – The patches need to be applied in the order that is specified in the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

- Reboot the system if necessary. Certain patches require a reboot to be effective.
x86 only: Rebooting the system is required or Live Upgrade fails.
`# init 6`

2 Create a ZFS root pool.

The ZFS root pool must be on a single slice to be bootable and upgradeable.

```
# zpool create rpool c3t0d0s0
```

In this example, the name of the new ZFS to be created is *rpool*. The pool is created on a bootable slice, *c3t0d0s0*.

For information about creating a new root pool, see the [Oracle Solaris ZFS Administration Guide](#).

3 Migrate your USF root (/) file system to the new ZFS root pool.

```
# lucreate [-c ufsBE] -n new-zfsBE -p rpool
```

- ufsBE* The name for the current UFS boot environment. This option is not required and is used only when the first boot environment is created. If you run the `lucreate` command for the first time and you omit the `-c` option, the software creates a default name for you.
- new-zfsBE* The name for the boot environment to be created. The name must be unique on the system.
- p rpool* Places the newly created ZFS root (/) file system into the ZFS root pool defined in *rpool*.

All nonshared non-global zones are copied to the new boot environment along with critical file systems. The creation of the new ZFS boot environment might take a while. The UFS file system data is being copied to the ZFS root pool. When the inactive boot environment has been created, you can use the `luupgrade` or `luactivate` command to upgrade or activate the new ZFS boot environment.

4 (Optional) Verify that the boot environment is complete.

The `lustatus` command reports whether the boot environment creation is complete and bootable.

```
# lustatus
boot environment   Is      Active   Active   Can      Copy
Name              Complete Now      OnReboot Delete   Status
-----
```

ufsBE	yes	yes	yes	no	-
new-zfsBE	yes	no	no	yes	-

5 (Optional) Verify the basic dataset information on the system.

The `list` command displays the names of all datasets on the system. In this example, `rpool` is the name of the ZFS pool and `new-zfsBE` is the name of the newly created ZFS boot environment.

```
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               11.4G  2.95G   31K    /rpool
rpool/ROOT                          4.34G  2.95G   31K    legacy
rpool/ROOT/new-zfsBE                4.34G  2.95G  4.34G    /
rpool/dump                          2.06G  5.02G   16K    -
rpool/swap                          5.04G  7.99G   16K    -
```

The mount points listed for the new boot environment are temporary until the `luactivate` command is executed. The `/dump` and `/swap` volumes are not shared with the original UFS boot environment, but are shared within the ZFS root pool and boot environments within the root pool.

Example 13–1 Migrating From a UFS root (/) File System With Non-Global Zones Installed to ZFS Root Pool

In the following example, the existing non-global zone `myzone`, has its non-global zone root in a UFS root (/) file system. The zone `zzone` has its zone root in a ZFS file system in the existing ZFS storage pool, `pool`. Live Upgrade is used to migrate the UFS boot environment, `c2t2d0s0`, to a ZFS boot environment, `zfs2BE`. The UFS-based `myzone` zone migrates to a new ZFS storage pool, `mpool`, that is created before the Live Upgrade operation. The ZFS-based, non-global zone, `zzone`, is cloned but retained in the ZFS pool `pool` and migrated to the new `zfs2BE` boot environment.

1. The commands to create the boot environment are as follows:

```
# zoneadm list -iv
ID NAME          STATUS    PATH                      BRAND  IP
0 global         running   /                        native shared
- myzone         installed /zones/myzone           native shared
- zzone          installed /pool/zones              native shared

# zpool create mpool mirror c3t0d0s0 c4td0s0
# lucreate -c c1t2d0s0 -n zfs2BE -p mpool
Checking GRUB menu...
Analyzing system configuration.
Updating boot environment description database on all BEs.
Updating system configuration files.
The device </dev/dsk/clt1d0s0> is not a root device for any boot environment; cannot get BE ID.
Creating configuration for boot environment <zfs2BE>.
Source boot environment is <c1t2d0s0>.
Creating file systems on boot environment <zfs2BE>.
Creating <zfs> file system for </> in zone <global> on <mpool/ROOT/zfs2BE>.
Populating file systems on boot environment <zfs2BE>.
```

Analyzing zones.
Mounting ABE <zfs2BE>.
Generating file list.
Copying data from PBE <clt2d0s0> to ABE <zfs2BE>.
100% of filenames transferred
Finalizing ABE.
Fixing zonepaths in ABE.
Unmounting ABE <zfs2BE>.
Fixing properties on ZFS datasets in ABE.
Reverting state of zones in PBE <clt2d0s0>.
Making boot environment <zfs2BE> bootable.
Updating bootenv.rc on ABE <zfs2BE>.
Saving existing file </boot/grub/menu.lst> in top level dataset for BE <zfs2BE> as <mount-point>//boot/grub/menu.lst.p
File </boot/grub/menu.lst> propagation successful
Copied GRUB menu from PBE to ABE
No entry for BE <zfs2BE> in GRUB menu
Population of boot environment <zfs2BE> successful.
Creation of boot environment <zfs2BE> successful.

2. When the lucreate operation completes, use the lustatus command to view the boot environment status as in this example.

```
# lustatus
Boot Environment      Is      Active Active      Can      Copy
Name                  Complete Now    On Reboot Delete Status
-----
clt2d0s0              yes     yes    yes         no        -
zfsBE                 yes     no     no          yes        -

# zoneadm list -iv
ID NAME                STATUS    PATH                                BRAND  IP
0 global              running   /                                    native shared
- myzone               installed /zones/myzone                      native shared
- zzone               installed /pool/zones                        native shared
```

3. Next, use the luactivate command to activate the new ZFS boot environment. For example:

```
# luactivate zfsBE
A Live Upgrade Sync operation will be performed on startup of boot
environment <zfsBE>.

*****

The target boot environment has been activated. It will be used when you
reboot. NOTE: You MUST NOT USE the reboot, halt, or uadmin commands. You
MUST USE either the init or the shutdown command when you reboot. If you
do not use either init or shutdown, the system will not boot using the
target BE.

*****

In case of a failure while booting to the target BE, the following
process
needs to be followed to fallback to the currently working boot
environment:
```

1. Enter the PROM monitor (ok prompt).

2. Boot the machine to Single User mode using a different boot device (like the Solaris Install CD or Network). Examples:

```
At the PROM monitor (ok prompt):
For boot to Solaris CD:  boot cdrom -s
For boot to network:    boot net -s
```

3. Mount the Current boot environment root slice to some directory (like /mnt). You can use the following command to mount:

```
mount -Fufs /dev/dsk/clt0d0s0 /mnt
```

4. Run <luactivate> utility with out any arguments from the current boot environment root slice, as shown below:

```
/mnt/sbin/luactivate
```

5. luactivate, activates the previous working boot environment and indicates the result.

6. Exit Single User mode and reboot the machine.

```
*****
```

```
Modifying boot archive service
Activation of boot environment <zfsBE> successful.
```

4. Reboot the system to the ZFS BE.

```
# init 6
# svc.startd: The system is coming down. Please wait.
# svc.startd: 79 system services are now being stopped.
.
.
.
```

5. Confirm the new boot environment and the status of the migrated zones as in this example.

```
# lustatus
Boot Environment      Is      Active Active   Can   Copy
Name                 Complete Now    On Reboot Delete Status
-----
clt2d0s0              yes     yes   yes     no    -
zfsBE                  yes     no    no      yes   -
```

If you fall back to the UFS boot environment, then you again need to import any ZFS storage pools that were created in the ZFS boot environment because they are not automatically available in the UFS boot environment. You will see messages similar to the following when you switch back to the UFS boot environment.

```
# luactivate clt2d0s0
WARNING: The following files have changed on both the current boot
environment <ZFSbe> zone <global> and the boot environment to be activated <clt2d0s0>:
/etc/zfs/zpool.cache
INFORMATION: The files listed above are in conflict between the current
boot environment <ZFSbe> zone <global> and the boot environment to be
activated <clt2d0s0>. These files will not be automatically synchronized
```

from the current boot environment <ZFSbe> when boot environment <c1t2d0s0>

PART III

Appendices

This part provides reference information.

Live Upgrade Command Reference

This appendix lists the commands that you can type at the command line. Live Upgrade includes man pages for all the listed command-line utilities, some of which are also described in this manual.

Task	Command
Activate an inactive boot environment.	luactivate(1M)
Cancel a scheduled copy or create job.	lucancel(1M)
Compare an active boot environment with an inactive boot environment.	lucompare(1M)
Recopy file systems to update an inactive boot environment.	lumake(1M)
Create a boot environment.	lucreate(1M)
Name the active boot environment.	lucurr(1M)
Delete a boot environment.	ludelete(1M)
Add a description to a boot environment name.	ludesc(1M)
List critical file systems for each boot environment.	lufslslist(1M)
Enable a mount of all of the file systems in a boot environment. This command enables you to modify the files in a boot environment while that boot environment is inactive.	lumount(1M)
Rename a boot environment.	lurename(1M)
List status of all boot environments.	lustatus(1M)

Task	Command
Enable an unmount of all the file systems in a boot environment. This command enables you to modify the files in a boot environment while that boot environment is inactive.	<code>luumount(1M)</code>
Upgrade an OS or install a Flash archive on an inactive boot environment.	<code>luupgrade(1M)</code>

Troubleshooting (Tasks)

This chapter contains a list of specific error messages and general problems you might encounter when installing the Oracle Solaris 10 8/11 OS. The chapter also explains how to fix the problems. The content is organized according to where in the installation process the problem occurred.

- “Problems With Setting Up Network Installations” on page 211
- “Problems With Booting a System” on page 212
- “Initial Installation of the Oracle Solaris OS” on page 217
- “Upgrading the Oracle Solaris OS” on page 220

Note – When you see the phrase “bootable media,” this means the Oracle Solaris installation program and JumpStart, a feature of Oracle Solaris, installation method.

Problems With Setting Up Network Installations

Unknown client “*host-name*”

Cause: The *host-name* argument in the `add_install_client` command is not a host in the naming service.

Solution: Add the host *host_name* to the naming service and execute the `add_install_client` command again.

Error: <system name> does not exist in the NIS ethers map

Add it, and rerun the `add_install_client` command

Description: When you run the `add_install_client` command, the command fails with the above error.

Cause: The client you are adding to the install server does not exist in the server's `/etc/ethers` file.

Solution: Add the needed information to the `/etc/ethers` file on the install server and run the `add_install_client` command again.

1. Become superuser or assume an equivalent role.

Note – Roles contain authorizations and privileged commands. For more information about roles, see “[Configuring RBAC \(Task Map\)](#)” in *System Administration Guide: Security Services*.

2. On the client, find the `ethers` address.

```
# ifconfig -a grep ethers
ether 8:0:20:b3:39:1d
```

3. On the install server, add the address to the list in the `/etc/ethers`.
4. On the client, run `add_install_client` again as in this example.

```
# ./add_install_client bluegill sun4u
```

Problems With Booting a System

Error Messages When Booting From Media

le0: No carrier - transceiver cable problem

Cause: The system is not connected to the network.

Solution: If the system is a nonnetworked system, ignore this message. If the system is a networked system, ensure that the Ethernet cabling is attached securely.

The file just loaded does not appear to be executable

Cause: The system cannot find the proper media for booting.

Solution: Verify that the system has been set up properly to install the Oracle Solaris 10 8/11 software from the network from an install server.

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

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

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

Solution: Try one of the following:

- Reset the boot - file in the PROM to ‘ ‘ (blank).
- Ensure that diag - switch is set to off and to true.

Can't boot from file/device

Cause: The installation media cannot find the bootable media.

Solution: Ensure that the following conditions are met:

- The DVD-ROM or CD-ROM drive is installed properly and turned on.
- Oracle Solaris Operating System DVD or the Oracle Solaris Software - 1 CD is inserted into the drive.
- The disc is free of damage or dirt.

WARNING: clock gained xxx days -- CHECK AND RESET DATE! (**SPARC based systems only**)

Description: This is an informational message.

Solution: Ignore the message and continue with the installation.

Not a UFS file system (**x86 based systems only**)

Cause: When the Oracle Solaris 10 8/11 software was installed (either through the Oracle Solaris installation program or custom JumpStart), no boot disk was selected. You now must edit the BIOS to boot the system.

Solution: Select the BIOS to boot. See your BIOS documentation for instructions.

General Problems When Booting From Media

The system does not boot.

Description: When initially setting up a JumpStart server, you might encounter boot problems that do not return an error message. To verify information about the system and how the system is booting, run the boot command with the -v option, which displays verbose debugging information.

Note – If you do not include this option, the messages are still printed, but the output is directed to the system log file. For more information, see the [syslogd\(1M\)](#) man page.

Solution: For SPARC based systems, at the ok prompt, type the following command:

```
ok boot net -v - install
```

Boot from DVD media fails on systems with Toshiba SD-M 1401 DVD-ROM

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

Solution: Apply patch 111649-03, or later version, to update the Toshiba SD-M1401 DVD-ROM drive's firmware. The patch 111649-03 is available at <http://support.oracle.com> (My Oracle Support) in the Patches and Updates tab.

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

Cause: Nonmemory PC cards cannot use the same memory resources that are used by other devices.

Solution: To correct this problem, see the instructions for your PC card and check for the address range.

The system hangs before displaying the system prompt. (**x86 based systems only**)

Cause: You have hardware that is not supported.

Solution: Check your hardware manufacturer's documentation.

Booting From the Network, Error Messages

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

Description: This error occurs when you have two or more servers on a network responding to an install client's boot request. The install client connects to the wrong boot server, and the installation hangs. The following specific reasons might cause this error to occur:

Cause: *Reason 1:* /etc/bootparams files might exist on different servers with an entry for this install client.

Solution: *Reason 1:* Ensure that servers on the network do not have multiple /etc/bootparams entries for the install client. If they do have multiple entries, remove duplicate client entries in the /etc/bootparams file on all install servers and boot servers except the one you want the install client to use.

Cause: *Reason 2:* Multiple /tftpboot or /rplboot directory entries might exist for this install client.

Solution: *Reason 2:* Ensure that servers on the network do not have multiple /tftpboot or /rplboot directory entries for the install client. If they do have multiple entries, remove duplicate client entries from the /tftpboot or /rplboot directories on all install servers and boot servers except the one you want the install client to use.

Cause: *Reason 3:* An install client entry might exist in the `/etc/bootparams` file on a server and an entry in another `/etc/bootparams` file that enables all systems to access the profile server. The entry would resemble the following:

```
* install_config=profile-server:path
```

A line that resembles the previous entry in the NIS or NIS+ bootparams table can also cause this error.

Solution: *Reason 3:* If a wildcard entry is in the naming service bootparams map or table (for example, `* install_config=`), delete it and add it to the `/etc/bootparams` file on the boot server.

No network boot server. Unable to install the system. See installation instructions. (**SPARC based systems only**)

Cause: A system on which you are attempting to install from the network is not set up correctly.

Solution: Ensure that you correctly set up the system to install from the network. See [“Adding Systems to Be Installed From the Network With a CD Image” in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations](#).

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

Cause: You are installing Oracle Solaris from a network but the boot software cannot locate the following:

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

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

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

Timeout waiting for ARP/RARP packet... (**SPARC based systems only**)

Cause: *Reason 1:* The client is trying to boot from the network but it cannot find a system that knows about the client.

Solution: *Reason 1:* Verify the system's host name is in the NIS or NIS+ naming service. Also, verify the bootparams search order in the boot server's `/etc/nsswitch.conf` file.

For example, the following line in the `/etc/nsswitch.conf` file indicates that JumpStart or the Oracle Solaris installation program first looks in the NIS maps for bootparams information. If the program does not find any information, the installer looks in the boot server's `/etc/bootparams` file.

```
bootparams: nis files
```

Cause: *Reason 2:* The client's Ethernet address is not correct.

Solution: *Reason 2:* Verify that the client's Ethernet address in the install server's `/etc/ethers` file is correct.

Cause: *Reason 3:* In a JumpStart installation, the `add_install_client` command specifies the platform group that uses a specified server as an install server. This problem occurs if the wrong architecture value is used when using the `add_install_client`. For example, the machine you want to install is a sun4u, but you used i86pc instead.

Solution: *Reason 3:* Rerun `add_install_client` with the correct architecture value.

`ip: joining multicasts failed on tr0 - will use link layer broadcasts for multicast (x86 based systems only)`

Cause: This error message is displayed when you boot a system with a token ring card. Ethernet multicast and token ring multicast do not work the same way. The driver returns this error message because an invalid multicast address was provided to it.

Solution: Ignore this error message. If multicast does not work, IP uses layer broadcasts instead and does not cause the installation to fail.

Requesting Internet address for *Ethernet-Address* (x86 based systems only)

Cause: The client is trying to boot from the network but it cannot find a system that knows about the client.

Solution: Verify the system's host name is listed in the naming service. If the system's host name is listed in the NIS or NIS+ naming service and the system continues to print this error message, try rebooting.

`RPC: Timed out No bootparams (whoami) server responding; still trying... (x86 based systems only)`

Cause: The client is trying to boot from the network but it cannot find a system with an entry in the `/etc/bootparams` file on the install server.

Solution: Use `add_install_client` on the install server to add the proper entry in the `/etc/bootparams` file, enabling the client to boot from the network.

Still trying to find a RPL server... (x86 based systems only)

Cause: The system is trying to boot from the network but the server is not set up to boot this system.

Solution: On the install server, execute `add_install_client` for the system to be installed. The `add_install_client` command sets up an `/rplboot` directory, which contains the necessary network boot program.

CLIENT MAC ADDR: FF FF FF FF FF FF (**network installations with DHCP only**)

Cause: The DHCP server is not configured correctly. This error might occur if the options or macros are not correctly defined in the DHCP Manager software.

Solution: In the DHCP Manager software, verify that the options and macros are correctly defined. Confirm that the Router option is defined, and that the value of the Router option is correct for the subnet you are using for the network installation.

General Problems When Booting From the Network

The system boots from the network, but from a system other than the specified install server.

Cause: An `/etc/bootparams` and perhaps an `/etc/ethers` entry exist on another system for the client.

Solution: On the name server, update the `/etc/bootparams` entry for the system that is being installed. The entry should conform to the following syntax:

```
install-system root=boot-server:path install=install-server:path
```

Also, ensure that only one `bootparams` entry is on the subnet for the install client.

The system does not boot from the network (**network installations with DHCP only**).

Cause: The DHCP server is not configured correctly. This error might occur if the system is not configured as an installation client on the DHCP server.

Solution: In the DHCP manager software, verify that installation options and macros are defined for the client system. For more information, see [“Preconfiguring System Configuration Information With the DHCP Service \(Tasks\)” in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations](#).

Initial Installation of the Oracle Solaris OS

Initial installation fails

Solution: If the Oracle Solaris installation fails, you must restart the installation. To restart the installation, boot the system from the Oracle Solaris Operating System DVD, the Oracle Solaris Software - 1 CD, or from the network.

You cannot uninstall the Oracle Solaris software after the software has been partially installed. You must restore your system from a backup or begin the Oracle Solaris installation process again.

/cdrom/cdrom0/SUNWxxx/reloc.cpio: Broken pipe

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

Solution: Ignore the message and continue with the installation.

WARNING: CHANGE DEFAULT BOOT DEVICE (x86 based systems only)

Cause: This is an informational message. The default boot device set in the system's BIOS might be set to a device that requires you to use the Oracle Solaris Device Configuration Assistant to boot the system.

Solution: Continue with the installation and, if necessary, change the system's default boot device specified in the BIOS after you install the Oracle Solaris software to a device that does not require the Oracle Solaris Device Configuration Assistant.

x86 only – If you are using the `locale` keyword to test a JumpStart profile for an initial installation, the `pfinstall -D` command fails to test the profile. For a workaround, see the error message “could not select locale,” in the section, “[Upgrading the Oracle Solaris OS](#)” on page 220.

▼ x86: How to Check an IDE Disk for Bad Blocks

IDE disk drives do not automatically map out bad blocks like other drives supported by Oracle Solaris software. Before installing Oracle Solaris on an IDE disk, you might want to perform a surface analysis on the disk.

1 Become superuser or assume equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see *Configuring RBAC (Task Map) in System Administration Guide: Security Services*.

2 Boot to the installation media.

3 When you are prompted to select an installation type, select option 6, Single user shell.

4 Start the `format(1M)` program.

```
# format
```

5 Specify the IDE disk drive on which you want to perform a surface analysis.

```
# cxdy
```

```
cx      The controller number
```

dy The device number

6 Determine whether you have an fdisk partition.

If an Oracle Solaris fdisk partition does not exist, use the fdisk command to create one on the disk.

```
format> fdisk
```

7 Begin the surface analysis.

```
format> analyze
```

8 Determine the current settings.

```
analyze> config
```

9 (Optional) Change settings.

```
analyze> setup
```

10 Determine whether any bad blocks exist.

```
analyze> type-of-surface-analysis
```

type-of-surface-analysis Read, write, or compare

If format finds bad blocks, it remaps them.

11 Exit the analysis.

```
analyze> quit
```

12 Specify blocks to remap: if necessary.

```
format> repair
```

13 Exit the format program.

```
quit
```

14 Restart the media in multiuser mode.

```
# exit
```

Upgrading the Oracle Solaris OS

Upgrading Error Messages

No upgradable disks

Cause: A swap entry in the `/etc/vfstab` file is causing the upgrade to fail.

Solution: Comment out the following lines in the `/etc/vfstab` file:

- All swap files and slices on disks not being upgraded
- Swap files that are no longer present
- Any unused swap slices

`usr/bin/bzcat` not found

Cause: Live Upgrade fails because of needing a patch cluster.

Solution: A patch is needed to install Live Upgrade. Ensure that you have the most recently updated patch list by consulting <http://support.oracle.com> (My Oracle Support). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

Upgradeable Solaris root devices were found, however, no suitable partitions to hold the Solaris install software were found. Upgrading using the Solaris Installer is not possible. It might be possible to upgrade using the Solaris Software 1 CDRom. (x86 based systems only)

Cause: You cannot upgrade with the Oracle Solaris Software - 1 CD because you do not have enough space.

Solution: To upgrade, you can either create a swap slice that is larger than or equal to 512 MB or use another method of upgrading such as the Oracle Solaris installation from Oracle Solaris Operating System DVD, a net installation image, or JumpStart.

ERROR: Could not select locale (**x86 based systems only**)

Cause: When you test your JumpStart profile by using the `pfinstall -D` command, the dry run test fails under the following conditions:

- The profile contains the locale keyword.
- You're testing a release that contains GRUB software. Starting with the Solaris 10 1/06 release, the GRUB boot loader facilitates booting different operating systems installed on your system with the GRUB menu.

With the introduction of GRUB software, the miniroot is compressed. The software can no longer find the list of locales from the compressed miniroot. The miniroot is the smallest possible Oracle Solaris root (/) file system and is found on the Oracle Solaris installation media.

Solution: Perform the following steps. Use the following values.

- MEDIA_DIR is /cdrom/cdrom0/
 - MINIROOT_DIR is \$MEDIA_DIR/Solaris_10/Tools/Boot
 - MINIROOT_ARCHIVE is \$MEDIA_DIR/boot/x86.miniroot
 - TEMP_FILE_NAME is /tmp/test
1. Become superuser or assume an equivalent role.
Roles contain authorizations and privileged commands. For more information about roles, see [“Configuring RBAC \(Task Map\)” in System Administration Guide: Security Services](#).
 2. Uncompress the miniroot archive.

```
# /usr/bin/gzcat $MINIROOT_ARCHIVE > $TEMP_FILE_NAME
```
 3. Create the miniroot device by using the lofiadm command.

```
# LOFI_DEVICE=/usr/sbin/lofiadm -a $TEMP_FILE_NAME
# echo $LOFI_DEVICE
/dev/lofi/1
```
 4. Mount the miniroot with the lofi command under the Miniroot directory.

```
# /usr/sbin/mount -F ufs $LOFI_DEVICE $MINIROOT_DIR
```
 5. Test the profile.

```
# /usr/sbin/install.d/pfinstall -D -c $MEDIA_DIR $path-to-jumpstart_profile
```
 6. After the testing is completed, unmount the lofi device.

```
# umount $LOFI_DEVICE
```
 7. Delete the lofi device.

```
# lofiadm -d $TEMP_FILE_NAME
```

General Problems When Upgrading

The upgrade option is not presented even though there is a version of Solaris software that’s upgradable on the system.

Cause: *Reason 1:* The /var/sadm directory is a symlink or it is mounted from another file system.

Solution: *Reason 1:* Move the /var/sadm directory into the root (/) or /var file system.

Cause: *Reason 2:* The `/var/sadm/softinfo/INST_RELEASE` file is missing.

Solution: *Reason 2:* Create a new `INST_RELEASE` file by using the following template:

```
OS=Solaris
VERSION=x
REV=0
```

`x` The version of Oracle Solaris software on the system

Cause: *Reason 3:* The `SUNWusr` package is missing from `/var/sadm/softinfo`.

Solution: *Solution 3:* You need to do an initial installation. The Oracle Solaris software is not upgradable.

Couldn't shut down or initialize the md driver

Solution: Follow these instructions:

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

The upgrade fails because the Solaris installation program cannot mount a file system.

Cause: During an upgrade, the script attempts to mount all the file systems that are listed in the system's `/etc/vfstab` file on the root (`/`) file system that is being upgraded. If the installation script cannot mount a file system, it fails and exits.

Solution: Ensure that all file systems in the system's `/etc/vfstab` file can be mounted. Comment out any file systems in the `/etc/vfstab` file that cannot be mounted or that might cause the problem so that the Oracle Solaris installation program does not try to mount them during the upgrade. Any system-based file systems that contain software to be upgraded (for example, `/usr`) cannot be commented out.

The upgrade fails

Description: The system does not have enough space for the upgrade.

Cause: Check [“Upgrading With Disk Space Reallocation” in *Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade*](#) for the space requirements and see if you can fix this issue without using auto-layout to reallocate space.

Problems upgrading RAID-1 volume root (`/`) file systems

Solution: If you have problems upgrading when using Solaris Volume Manager RAID-1 volumes that are the root (`/`) file system, see [Chapter 25, “Troubleshooting Solaris Volume Manager \(Tasks\)” in *Solaris Volume Manager Administration Guide*](#).

▼ How to Continue Upgrading After a Failed Upgrade

If the upgrade fails and the system cannot be soft-booted for reasons beyond your control, such as a power failure or a network connection failure, try to continue upgrading.

- 1 **Reboot the system from the Oracle Solaris Operating System DVD, the Oracle Solaris Software - 1 CD, or from the network.**
- 2 **Choose the upgrade option for installation.**

The Oracle Solaris installation program determines whether the system has been partially upgraded and continues the upgrade.

x86: Problems With Live Upgrade When You Use GRUB

The following errors can occur when you use Live Upgrade and the GRUB boot loader on an x86 based system.

ERROR: The media product tools installation directory *path-to-installation-directory* does not exist.

ERROR: The media *dirctory* does not contain an operating system upgrade image.

Description: These error messages can occur when using the `luupgrade` command to upgrade a new boot environment.

Cause: An older version of Live Upgrade is being used. The Live Upgrade packages you have installed on your system are incompatible with the media and the release on that media.

Solution: Always use the Live Upgrade packages from the release you are upgrading to.

Example: In the following example, the error message indicates that the Live Upgrade packages on the system are not the same version as on the media.

```
# luupgrade -u -n s10u1 -s /mnt
Validating the contents of the media </mnt>.
The media is a standard Solaris media.
ERROR: The media product tools installation directory
</mnt/Solaris_10/Tools/Boot/usr/sbin/install.d/install_config> does
not exist.
ERROR: The media </mnt> does not contain an operating system upgrade
image.
```

ERROR: Cannot find or is not executable: </sbin/biosdev>.

ERROR: One or more patches required by Live Upgrade has not been installed.

Cause: One or more patches required by Live Upgrade are not installed on your system. Beware that this error message does not catch all missing patches.

Solution: Before using Live Upgrade, always install all the required patches. Ensure that you have the most recently updated patch list by consulting <http://support.oracle.com> (My Oracle Support). Search for the knowledge document 1004881.1 - Live Upgrade Software Patch Requirements (formerly 206844) on My Oracle Support.

ERROR: Device mapping command </sbin/biosdev> failed. Please reboot and try again.

Cause: *Reason 1:* Live Upgrade is unable to map devices because of previous administrative tasks.

Solution: *Reason 1:* Reboot the system and try Live Upgrade again

Cause: *Reason 2:* If you reboot your system and get the same error message, you have two or more identical disks. The device mapping command is unable to distinguish between them.

Solution: *Reason 2:* Create a new dummy fdisk partition on one of the disks and then reboot the system. For more information, see the [fdisk\(1M\)](#) man page.

Cannot delete the boot environment that contains the GRUB menu

Cause: Live Upgrade imposes the restriction that a boot environment cannot be deleted if the boot environment contains the GRUB menu.

Solution: Use the [lumake\(1M\)](#) or [luupgrade\(1M\)](#) commands to reuse that boot environment.

The file system containing the GRUB menu was accidentally remade. However, the disk has the same slices as before. For example, the disk was not re-sliced.

Cause: The file system that contains the GRUB menu is critical to keeping the system bootable. Live Upgrade commands do not destroy the GRUB menu. But, if you accidentally remake or otherwise destroy the file system containing the GRUB menu with a command other than a Live Upgrade command, the recovery software attempts to reinstall the GRUB menu. The recovery software puts the GRUB menu back in the same file system at the next reboot. For example, you might have used the `newfs` or `mkfs` commands on the file system and accidentally destroyed the GRUB menu. To restore the GRUB menu correctly, the slice must adhere to the following conditions:

- Contains a mountable file system
- Remains a part of the same Live Upgrade boot environment where the slice resided previously

Before rebooting the system, make any necessary corrective actions on the slice.

Solution: Reboot the system. A backup copy of the GRUB menu is automatically installed.

The GRUB menu's `menu.lst` file was accidentally deleted.

Solution: Reboot the system. A backup copy of the GRUB menu is automatically installed.

System Panics When Upgrading With Live Upgrade Running Veritas VxVm

▼ How to Upgrade When Running Veritas VxVm

When you use 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 Oracle Solaris advanced packaging guidelines.

- 1 **Become superuser or assume an equivalent role.**
Roles contain authorizations and privileged commands. For more information about roles, see *Configuring RBAC (Task Map) in System Administration Guide: Security Services*.
- 2 **Create an inactive boot environment. See [“Creating a New Boot Environment” on page 64](#).**
- 3 **Before upgrading the inactive boot environment, 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 /mnt
```

- b. **Change to the directory that contains the `vfstab` file.**

For example:

```
# cd /mnt/etc
```

- c. **Make a copy of the inactive boot environment's `vfstab` file.**

For example:

```
# cp vfstab vfstab.501
```

- d. **In the copied `vfstab`, comment out all Veritas file system entries.**

For example:

```
# sed '/vx\/dsk\/s\/^\/#/g' < vfstab > vfstab.novxfs
```

The first character of each line is changed to #, which makes the line a comment line. Note that this comment line is different than the system file-comment lines.

e. Copy the changed `vfstab` file.

For example:

```
# cp vfstab.novxfs vfstab
```

f. Change directories to the inactive boot environment's system file.

For example:

```
# cd /mnt/etc
```

g. Make a copy of the inactive boot environment's system file.

For example:

```
# cp system system.501
```

h. Comment out all “`forceload:`” entries that include `drv/vx`.

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

The first character of each line is changed to `*`, which makes the line a command line. Note that this comment line is different than the `vfstab` file comment lines.

i. Create the Veritas `install-db` file.

For example:

```
# touch vx/reconfig.d/state.d/install-db
```

j. Unmount the inactive boot environment.

```
# luumount inactive-boot-environment-name
```

4 Upgrade the inactive boot environment.

See [Chapter 5, “Upgrading With Live Upgrade \(Tasks\).”](#)

5 Activate the inactive boot environment.

See “[Activating a Boot Environment](#)” on page 104.

6 Shut down the system.

```
# init 0
```

7 Boot the inactive boot environment in single-user mode.

```
OK boot -s
```

Several messages and error messages that contain “`vxvm`” or “`VXVM`” are displayed that can be ignored. The inactive boot environment becomes active.

8 Upgrade Veritas.

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

```
# pkgrm VRTSvmsa
```

- b. Change directories to the Veritas packages.

```
# cd /location-of-Veritas-software
```

- c. Add the latest Veritas packages to the system.

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

9 Restore the original vfstab and system files.

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

10 Reboot the system.

```
# init 6
```

x86: Service Partition Not Created by Default on Systems With No Existing Service Partition

If you install the Oracle Solaris 10 8/11 OS on a system that does not currently include a service or diagnostic partition, the installation program might not create a service partition by default. If you want to include a service partition on the same disk as the Oracle Solaris partition, you must re-create the service partition before you install the Oracle Solaris 10 8/11 OS.

If you installed the Solaris 8 2/02 OS on a system with a service partition, the installation program might not have preserved the service partition. If you did not manually edit the fdisk boot partition layout to preserve the service partition, the installation program deleted the service partition during the installation.

Note – If you did not specifically preserve the service partition when you installed the Solaris 8 2/02 OS, you might not be able to re-create the service partition and upgrade to the Oracle Solaris 10 8/11 OS.

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

▼ **How to Include a Service Partition When Installing Software From a Network Installation Image or From the Oracle Solaris Operating System DVD**

To install the software from a net installation image or from the Oracle Solaris Operating System DVD over the network, follow these steps.

- 1 Delete the contents of the disk.**
- 2 Before you install, create the service partition by using the diagnostics CD for your system.**
For information about how to create the service partition, see your hardware documentation.
- 3 Boot the system from the network.**
The Customize fdisk Partitions screen is displayed.
- 4 Load the default boot disk partition layout by clicking Default.**
The installation program preserves the service partition and creates the Oracle Solaris partition.

▼ **How to Include a Service Partition When Installing From the Oracle Solaris Software - 1 CD or From a Network Installation Image**

To use the Oracle Solaris installation program to install from the Oracle Solaris Software - 1 CD or from a network installation image on a boot server, follow these steps.

- 1 Delete the contents of the disk.**
- 2 Before you install, create the service partition by using the diagnostics CD for your system.**
The installation program prompts you to choose a method for creating the Oracle Solaris partition. For information about how to create the service partition, see your hardware documentation.
- 3 Boot the system.**
- 4 Select the Use rest of disk for Solaris partition option.**
The installation program preserves the service partition and creates the Oracle Solaris partition.
- 5 Complete the installation.**

Additional SVR4 Packaging Requirements (Reference)

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

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

This chapter contains the following sections:

- [“Preventing Modification of the Current OS” on page 229.](#)
- [“Preventing User Interaction When Installing or Upgrading” on page 233.](#)
- [“Setting Package Parameters For Zones” on page 234](#)

Preventing Modification of the Current OS

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

Using Absolute Paths

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

Packages can include absolute paths in their pkgmap file (package map). If these files exist, they are written relative to the -R option of the pkgadd command. Packages that contain both absolute and relative (relocatable) paths can be installed to an alternative root (/) file system as well. \$PKG_INSTALL_ROOT is prepended to both absolute and relocatable files so all paths are resolved properly when being installed by pkgadd.

Using the pkgadd -R Command

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

Any procedure scripts that are included in the packages being installed with the `pkgadd` command `-R` option or being removed by using the `pkgrm` command `-R` option must not alter the currently running system. Any installation scripts that you provide must reference any directory or file that is prefixed with the `$PKG_INSTALL_ROOT` variable. The package must write all directories and files with the `$PKG_INSTALL_ROOT` prefix. The package must not remove directories without a `$PKG_INSTALL_ROOT` prefix.

The following table provides examples of script syntax.

TABLE C-1 Examples of Installation Script Syntax

Script Type	Correct Syntax	Incorrect Syntax
Bourne shell “if” statement fragments	<code>if [-f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf] ; then</code>	<code>if [-f /etc/myproduct.conf] ; \ then</code>
Removing a file	<code>/bin/rm -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</code>	<code>/bin/rm -f /etc/myproduct.conf</code>
Changing a file	<code>echo "test=no" > \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</code>	<code>echo "test=no" > \ /etc/myproduct.conf</code>

Differences Between \$PKG_INSTALL_ROOT and \$BASEDIR

`$PKG_INSTALL_ROOT` is the location of the root (/) file system of the machine to which you are adding the package. The location is set to the `-R` argument of the `pkgadd` command. For example, if the following command is invoked, then `$PKG_INSTALL_ROOT` becomes `/a` during the installation of the package.

```
# pkgadd -R /a SUNWvxvm
```

`$BASEDIR` points to the *relocatable* base directory into which relocatable package objects are installed. Only relocatable objects are installed here. Nonrelocatable objects (those that have *absolute* paths in the `pkgmap` file) are always installed relative to the inactive boot environment, but not relative to the `$BASEDIR` in effect. If a package has no relocatable objects, then the package is said to be an absolute package (or nonrelocatable), and `$BASEDIR` is undefined and not available to package procedure scripts.

For example, suppose a package's `pkgmap` file has two entries:

```
1 f none sbin/ls 0555 root sys 3541 12322 1002918510
1 f none /sbin/ls2 0555 root sys 3541 12322 2342423332
```

The pkginfo file has a specification for \$BASEDIR:

```
BASEDIR=/opt
```

If this package is installed with the following command, then ls is installed in /a/opt/sbin/ls, but ls2 is installed as /a/sbin/ls2.

```
# pkgadd -R /a SUNWtest
```

Guidelines for Writing Scripts

Your package procedure scripts must be independent of the currently running OS to prevent modifying the OS. Procedure scripts define actions that occur at particular points during package installation and removal. Four procedure scripts can be created with these predefined names: preinstall, postinstall, preremove, and postremove.

TABLE C-2 Guidelines for Creating Scripts

Guidelines	Affects Live Upgrade	Affects non-global zones
Scripts must be written in Bourne shell (/bin/sh). Bourne shell is the interpreter that is used by the pkgadd command to execute the procedure scripts.	X	X
Scripts must not start or stop any processes or depend on the output of commands such as ps or truss, which are operating system dependent and report information about the currently running system.	X	X
Scripts are free to use other standard UNIX commands such as expr, cp, and ls and other commands that facilitate shell scripting.	X	X
Any commands that a script invokes must be available in all supported releases, because a package must run on all of those releases. Therefore, you cannot use commands that were added or removed after the Solaris 8 release.	X	
To verify that a specific command or option is supported in a Solaris 8, Solaris 9, or Oracle Solaris 10 release, see the specific Reference Manual version on http://www.oracle.com/technetwork/indexes/documentation/index.html .		

Maintaining Diskless Client Compatibility

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

Verifying Packages

All packages must pass `pkgchk` validation. After a package is created and before it is installed, it must be checked with the following command.

```
# pkgchk -d dir-name pkg-name
```

dir-name Specifies the name of the directory where the package resides

pkg-name Specifies the name of the package

EXAMPLE C-1 Testing a Package

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

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

No errors should be displayed.

EXAMPLE C-2 Testing a Package on /export/SUNWvxxm

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

```
# pkgchk -d /export SUNWvxxm
```

No errors should be displayed.

Other commands can check the package when you are creating, modifying, and deleting files. For example:

- The `dircmp` or `fssnap` commands can be used to verify that packages behave properly.
- The `ps` command can be used for testing daemon compliance by making sure daemons are not stopped or started by the package.
- The `truss`, `pkgadd -v`, and `pkgrm` commands can test runtime package installation compliance but might not work in all situations. In the following example, the `truss` command strips out all read-only, non-`$TMPDIR` access and shows only non-read-only access to paths that do not lie within the specified inactive boot environment.

```
# TMPDIR=/a; export TMPDIR
# truss -t open /usr/sbin/pkgadd -R ${TMPDIR} SUNWvxxm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v \
'open('${TMPDIR}
```


Preventing User Interaction When Installing or Upgrading

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

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

To test a package to ensure that it will install with no user interaction, a new administration file can be set up with the `pkgadd` command `-a` option. The `-a` option defines an installation administration file to be used in place of the default administration file. Using the default file might result in the user being prompted for more information. You can create an administration file that indicates to `pkgadd` that it should bypass these checks and install the package without user confirmation. For details, see the [admin\(4\)](#) or [pkgadd\(1M\)](#) man page.

The following examples show how the `pkgadd` command uses the administration file.

- If no administration file is provided, `pkgadd` uses `/var/sadm/install/admin/default`. Using this file might result in user interaction.


```
# pkgadd
```
- If a relative administration file is provided on the command line, `pkgadd` looks in `/var/sadm/install/admin` for the file name and uses it. In this example, the relative administration file is named `nocheck` and `pkgadd` looks for `/var/sadm/install/admin/nocheck`.


```
# pkgadd -a nocheck
```
- If an absolute file is provided `pkgadd` uses it. In this example, `pkgadd` looks in `/tmp` for the `nocheck` administration file.


```
# pkgadd -a /tmp/nocheck
```

EXAMPLE C-3 Installation Administration File

The following example shows an installation administration file that requires very little user interaction with the `pkgadd` utility. Unless the package requires more space than is available on the system, the `pkgadd` utility uses this file and installs the package without prompting the user for more information.

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
space=ask
setuid=nocheck
conflict=nocheck
```

EXAMPLE C-3 Installation Administration File (Continued)

```
action=nocheck
basedir=default
```

Setting Package Parameters For Zones

Packages have parameters that control how their content is distributed and made visible on a system with non-global zones installed. The SUNW_PKG_ALLZONES, SUNW_PKG_HOLLOW, and SUNW_PKG_THISZONE package parameters define the characteristics of packages on a system with zones installed. These parameters must be set so that packages can be administered in a system with non-global zones.

Table C-3 lists the four valid combinations for setting package parameters. If you choose setting combinations that are not listed in the table, those settings are invalid and result in the package failing to install.

Note – Ensure that you have set all three package parameters. You can leave all three package parameters blank. The package tools interpret a missing zone package parameter as if the setting were “false,” but not setting the parameters is strongly discouraged. By setting all three package parameters, you specify the exact behavior the package tools should exhibit when installing or removing the package.

TABLE C-3 Valid Package Parameter Settings For Zones

SUNW_PKG_ALLZONES Setting	SUNW_PKG_HOLLOW Setting	SUNW_PKG_THISZONE Setting	Package Description
false	false	false	<p>The default setting for packages that do not specify values for all the zone package parameters.</p> <p>A package with these settings can be installed in either the global zone or a non-global zone.</p> <ul style="list-style-type: none">■ If the pkgadd command is run in the global zone, the package is installed in the global zone and in all non-global zones.■ If the pkgadd command is run in a non-global zone, the package is installed in the non-global zone only. <p>In both cases, the entire contents of the package is visible in all zones where the package is installed.</p>

TABLE C-3 Valid Package Parameter Settings For Zones (Continued)

SUNW_PKG_ALLZONES Setting	SUNW_PKG_HOLLOW Setting	SUNW_PKG_THISZONE Setting	Package Description
false	false	true	<p>A package with these settings can be installed in either the global zone or a non-global zone. If new non-global zones are created after the installation, the package is not propagated to these new non-global zones.</p> <ul style="list-style-type: none"> ■ If the pkgadd command is run in the global zone, the package is installed in the global zone only. ■ If the pkgadd command is run in a non-global zone, the package is installed in the non-global zone only. <p>In both cases, the entire contents of the package is visible in the zone where the package is installed.</p>
true	false	false	<p>A package with these settings can be installed in the global zone only. When the pkgadd command is run, the package is installed in the global zone and in all non-global zones. The entire contents of the package is visible in all zones.</p> <p>Note – Any attempt to install the package in a non-global zone fails.</p>

TABLE C-3 Valid Package Parameter Settings For Zones (Continued)

SUNW_PKG_ALLZONES Setting	SUNW_PKG_HOLLOW Setting	SUNW_PKG_THISZONE Setting	Package Description
true	true	false	<p>A package with these settings can only be installed in the global zone, by the global administrator. When the pkgadd command is run, the contents of the package is fully installed in the global zone. If a package has the package parameters set to these values, the package content itself is not delivered on any non-global zone. Only the package installation information necessary to make the package appear to be installed is installed on all non-global zones. This enables the installation of other packages to be installed that depend on this package. For more information about “hollow” packages, see Chapter 25, “About Packages and Patches on an Oracle Solaris System With Zones Installed (Overview)”, in <i>System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones</i>.</p> <p>For package dependency checking purposes, the package appears to be installed in all zones.</p> <ul style="list-style-type: none">■ In the global zone, the entire contents of the package is visible.■ In whole root non-global zones, the entire contents of the package is not visible.■ When a non-global zone inherits a file system from the global zone, a package installed in this file system is visible in a non-global zone. All other files delivered by the package are not visible within the non-global zone. <p>For example, a sparse root non-global zone shares certain directories with the global zone. These directories are read-only. Sparse root non-global zones share the /platform file system among others. Another example is packages that deliver files relevant only to booting hardware.</p> <p>Note – Any attempt to install the package in a non-global zone fails.</p>

Description	For More Information
For more details about packages and zones	Chapter 25, “About Packages and Patches on an Oracle Solaris System With Zones Installed (Overview),” in <i>System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones</i>
For an overview of sparse and whole root zones	Chapter 16, “Introduction to Solaris Zones,” in <i>System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones</i>
For information about package characteristics and parameters	pkginfo(4) man page
For information about displaying package parameter values	pkgparam(1) man page

For Further Information

The following references provide background information about the topics in this appendix.

For more specific information about packaging requirements and definitions of terminology	Chapter 6, “Advanced Techniques for Creating Packages,” in <i>Application Packaging Developer’s Guide</i>
For basic information about adding and removing packages and the installation administration file	Chapter 20, “Managing Software (Overview),” in <i>Oracle Solaris Administration: Basic Administration</i>
For detailed information about specific commands that are referenced in this appendix, see these man pages	dircmp(1) , fssnap(1M) , ps(1) , or truss(1) pkgadd(1M) , pkgchk(1M) , or pkgrm(1M) man pages
For an overview of Live Upgrade	Chapter 2, “Live Upgrade (Overview)”
For an overview of JumpStart	Chapter 2, “JumpStart (Overview),” in <i>Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations</i>
For an overview of Oracle Solaris Zones	Chapter 16, “Introduction to Solaris Zones,” in <i>System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones</i>

Using the Patch Analyzer When Upgrading (Tasks)

This chapter provides instructions on checking patches with the Patch Analyzer before upgrading the Oracle Solaris OS. The Patch Analyzer performs an analysis on your system if you want to upgrade to one of the following:

- Solaris 10 1/06 release
- Solaris 10 6/06 release
- Solaris 10 11/06
- Solaris 10 8/07
- Solaris 10 5/08
- Solaris 10 10/08

Upgrading to an Oracle Solaris Update Release

If you are already running the Solaris 10 3/05 release OS and have installed individual patches, upgrading to a subsequent Oracle Solaris 10 release causes the following:

- Any patches that are supplied as part of one of the releases noted above are reapplied to your system. You cannot back out these patches.
- Any patches that were previously installed on your system that are not included in one of the releases noted above are removed.

The Patch Analyzer performs an analysis on your system to determine which patches, if any, will be removed by upgrading to any of the above releases. The Patch Analyzer is available in the following formats.

- If you are using the Oracle Solaris installation program to upgrade, the Patch Analyzer dialog box appears. Select Yes to perform the analysis.
- If you are using the text installer to upgrade, select Analyze in the Analyze Patches dialog box to perform the analysis.
- If you are using a JumpStart installation or Live Upgrade, a feature of Oracle Solaris, to upgrade, run the `analyze_patches` script to perform the analysis.

▼ How to Use the `analyze_patches` Script

Note – To run the `analyze_patches` script, the installed system and the Oracle Solaris Operating System DVD, Oracle Solaris Software CDs, or network installation image must be accessible by the script either through NFS or locally mounted media.

1 Change to the `Misc` directory.

In this example, the image is located on locally mounted media.

```
# cd /cdrom/cdrom0/Solaris_10/Misc
```

2 Run the `analyze_patches` script.

```
# ./analyze_patches -R rootdir -N netdir -D databasedir
```

`-R rootdir` *rootdir* is the root of the installed system. The default is `/`.

`-N netdir` *netdir* is the path to the root of the OS image to be installed. The default is `/cdrom/cdrom0`. *netdir* is the path to the directory that contains the `Solaris_10` directory. You must use this option if you are running the `patch_analyzer` from an NFS mount point.

`-D databasedir` If the script is invoked from a directory other than the `Misc/` directory in the OS image, the program cannot find the database it uses for patch analysis. Use the `-D` option to supply the path to the database. Without this database, which is located in `Solaris_10/Misc/database` on the OS image, the script does not work properly.

3 Review the output of the Patch Analyzer.

The Patch Analyzer provides a list of patches that will be removed, downgraded, accumulated, or obsoleted by other patches. Patch accumulations are similar to patch upgrades. The accumulated patch is removed and its fixes are delivered by a new patch. Messages such as the following are shown:

```
Patch 105644-03 will be removed.
```

```
Patch 105925 will be downgraded from -02 to -01.
```

```
Patch 105776-01 will be accumulated/obsoleted by patch 105181-05.
```

If the Patch Analyzer program does not provide a list, no action is taken against any patches that were previously installed on your system.

4 If the patch replacements and deletions are acceptable, upgrade the system.

Glossary

archive	<p>A file that contains a collection of files that were copied from a master system. The file also contains identification information about the archive, such as a name and the date that you created the archive. After you install an archive on a system, the system contains the exact configuration of the master system.</p> <p>An archive could be a differential archive, which is a flash archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files specified and is restricted to systems that contain software consistent with the unchanged master image.</p>
begin script	<p>A user-defined Bourne shell script, specified within the <code>rules</code> file, that performs tasks before the Oracle Solaris software is installed on the system. You can use begin scripts only with JumpStart installations.</p>
boot	<p>To load the system software into memory and start it.</p>
boot archive	<p>x86 only: A boot archive is a collection of critical files that is used to boot the Oracle Solaris OS. These files are needed during system startup before the root (/) file system is mounted. Two boot archives are maintained on a system:</p> <ul style="list-style-type: none">■ The boot archive that is used to boot the Oracle Solaris OS on a system. This boot archive is sometimes called the primary boot archive.■ The boot archive that is used for recovery when the primary boot archive is damaged. This boot archive starts the system without mounting the root (/) file system. On the GRUB menu, this boot archive is called failsafe. The archive's essential purpose is to regenerate the primary boot archive, which is usually used to boot the system.
boot environment	<p>A collection of mandatory file systems (disk slices and mount points) that are critical to the operation of the Oracle Solaris OS. These disk slices might be on the same disk or distributed across multiple disks.</p> <p>The active boot environment is the one that is currently booted. Exactly one active boot environment can be booted. An inactive boot environment is not currently booted, but can be in a state of waiting for activation on the next reboot.</p>
boot loader	<p>x86 only: The boot loader is the first software program that runs after you turn on a system. This program begins the booting process.</p>
boot server	<p>A server system that provides client systems on the same network subnet with the programs and information that they need to start. A boot server is required to install over the network if the install server is on a different subnet than the systems on which Oracle Solaris software is to be installed.</p>

Core Software Group	A software group that contains the minimum software that is required to boot and run the Oracle Solaris OS on a system. Core includes some networking software and the drivers that are required to run the Common Desktop Environment (CDE) desktop. Core does not include the CDE software.
critical file systems	File systems that are required by the Oracle Solaris OS. When you use Live Upgrade, a feature of Oracle Solaris, these file systems are separate mount points in the <code>vfstab</code> file of the active and inactive boot environments. Example file systems are <code>root (/)</code> , <code>/usr</code> , <code>/var</code> , and <code>/opt</code> . These file systems are always copied from the source to the inactive boot environment.
dataset	A generic name for the following ZFS entities: clones, file systems, snapshots, or volumes.
Developer Oracle Solaris Software Group	A software group that contains the End User Oracle Solaris Software Group plus the libraries, include files, man pages, and programming tools for developing software.
DHCP	(Dynamic Host Configuration Protocol) An application-layer protocol. Enables individual computers, or clients, on a TCP/IP network to extract an IP address and other network configuration information from a designated and centrally maintained DHCP server or servers. This facility reduces the overhead of maintaining and administering a large IP network.
End User Oracle Solaris Software Group	A software group that contains the Core Software Group plus the recommended software for an end user, including the Common Desktop Environment (CDE) and DeskSet software.
Entire Oracle Solaris Software Group	A software group that contains the entire Oracle Solaris release.
Entire Oracle Solaris Software Group Plus OEM Support	A software group that contains the entire Oracle Solaris release plus additional hardware support for OEMs. This software group is recommended when installing Oracle Solaris software on SPARC based servers.
/etc directory	A directory that contains critical system configuration files and maintenance commands.
/export file system	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 <code>root (/)</code> 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.
failsafe boot archive	x86 only: A boot archive that is used for recovery when the primary boot archive is damaged. This boot archive starts the system without mounting the <code>root (/)</code> file system. This boot archive is called failsafe on the GRUB menu. The archive's essential purpose is to regenerate the primary boot archive, which is usually used to boot the system. See <i>boot archive</i> .
fallback	A reversion to the environment that ran previously. Use fallback when you are activating an environment and the boot environment that is designated for booting fails or shows some undesirable behavior.

fdisk partition	A logical partition of a disk drive that is dedicated to a particular operating system on x86 based systems. To install the Oracle Solaris software, you must set up at least one Oracle Solaris <code>fdisk</code> partition on an x86 based system. x86 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 Oracle Solaris <code>fdisk</code> partition per disk.
file server	A server that provides the software and file storage for systems on a network.
file system	In the Oracle Solaris OS, a tree-structured network of files and directories that you can access.
finish script	A user-defined Bourne shell script, specified within the <code>rules</code> file, that performs tasks after the Oracle Oracle Solaris software is installed on the system but before the system reboots. You use finish scripts with JumpStart installations.
flash archive	An Oracle Solaris installation feature that enables you to create an archive of the files on a system, called the <i>master system</i> . You can then use the archive to install other systems, making the other systems identical in their configuration to the master system. See also <i>archive</i> .
format	To put data into a structure or divide a disk into sectors for receiving data.
global zone	In Oracle Solaris Zones, the global zone is both the default zone for the system and the zone used for system-wide administrative control. The global zone is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled. Administration of the system infrastructure, such as physical devices, routing, or dynamic reconfiguration (DR), is only possible in the global zone. Appropriately privileged processes running in the global zone can access objects associated with other zones. See also <i>Oracle Solaris Zones</i> and <i>non-global zone</i> .
GRUB	x86 only: GNU GRand Unified Bootloader (GRUB) is an open source boot loader with a simple menu interface. The menu displays a list of operating systems that are installed on a system. GRUB enables you to easily boot these various operating systems, such as the Oracle Solaris OS, Linux, or Microsoft Windows.
GRUB edit menu	x86 only: A boot menu that is a submenu of the GRUB main menu. GRUB commands are displayed on this menu. These commands can be edited to change boot behavior.
GRUB main menu	x86 only: A boot menu that lists the operating systems that are installed on a system. From this menu, you can easily boot an operating system without modifying the BIOS or <code>fdisk</code> partition settings.
initial installation	An installation that overwrites the currently running software or initializes a blank disk. An initial installation of the Oracle Solaris OS overwrites the system's disk or disks with the new version of the Oracle Solaris OS. If your system is not running the Oracle Solaris OS, you must perform an initial installation. If your system is running an upgradable version of the Oracle Solaris OS, an initial installation overwrites the disk and does not preserve the OS or local modifications.
install server	A server that provides the Oracle Solaris DVD or CD images from which other systems on a network can install Oracle Solaris (also called a <i>media server</i>). You can create an install server by copying the Oracle Solaris DVD or CD images to the server's hard disk.

JumpStart	A type of installation in which the Oracle Solaris software is automatically installed on a system that is based on a user-defined profile. You can create customized profiles for different types of users and systems. A custom JumpStart installation is a JumpStart installation you create.
JumpStart installation	A type of installation in which the Oracle Solaris software is automatically installed on a system by using the factory-installed JumpStart software.
Live Upgrade	An upgrade method that enables a duplicate boot environment to be upgraded while the active boot environment is still running, thus eliminating downtime of the production environment.
locale	A geographic or political region or community that shares the same language, customs, or cultural conventions (English for the U.S. is <code>en_US</code> , and English for the U.K. is <code>en_UK</code>).
menu.lst file	x86 only: A file that lists all the operating systems that are installed on a system. The contents of this file dictate the list of operating systems that is displayed on the GRUB menu. From the GRUB menu, you can easily boot an operating system without modifying the BIOS or <code>fdisk</code> partition settings.
miniroot	A minimal, bootable root (<code>/</code>) file system that is included in Oracle Solaris installation media. A miniroot consists of the Oracle Solaris software that is required to install and upgrade systems. On x86 based systems, the miniroot is copied to the system to be used as the failsafe boot archive. See <i>failsafe boot archive</i> .
mirror	See <i>RAID-1 volume</i> .
mount	The process of accessing a directory from a disk that is attached to a machine that is making the mount request or a remote disk on a network. To mount a file system, you need a mount point on the local system and the name of the file system to be mounted (for example, <code>/usr</code>).
mount point	A workstation directory to which you mount a file system that exists on a remote machine.
network installation	A way to install software over the network from a system with a CD-ROM or DVD-ROM drive to a system without a CD-ROM or DVD-ROM drive. Network installations require a <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 that they can communicate and share information. Referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.
NIS	The SunOS 4.0 (minimum) Network Information Service. A distributed network database that contains key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.
NIS+	The SunOS 5.0 (minimum) Network Information Service. NIS+ replaces NIS, the SunOS 4.0 (minimum) Network Information Service.
non-global zone	A virtualized operating system environment created within a single instance of the Oracle Solaris Operating System. One or more applications can run in a non-global zone without interacting with the rest of the system. Non-global zones are also called zones. See also <i>Oracle Solaris Zones</i> and <i>global zone</i> .
nonnetworked systems	Systems that are not connected to a network or do not rely on other systems.
/opt file system	A file system that contains the mount points for third-party and unbundled software.

Oracle Solaris DVD or CD images	The Oracle Solaris software that is installed on a system, which you can access on the Oracle Solaris DVDs or CDs or an install server's hard disk to which you have copied the Oracle Solaris DVD or CD images.
Oracle Solaris installation program	A graphical user interface (GUI) or command-line interface (CLI) installation program that uses wizard panels to guide you step-by-step through installing the Oracle Solaris software and third-party software.
Oracle Solaris Zones	A software partitioning technology used to virtualize operating system services and provide an isolated and secure environment for running applications. When you create a non-global zone, you produce an application execution environment in which processes are isolated from all other zones. This isolation prevents processes that are running in a zone from monitoring or affecting processes that are running in any other zones. See also <i>global zone</i> and <i>non-global zone</i> .
package	A collection of software that is grouped into a single entity for modular installation. The Oracle Solaris software is divided into <i>software groups</i> , which are each composed of <i>clusters</i> and packages.
patch analyzer	A script that you can run manually or as part of the Oracle Solaris installation program. The patch analyzer performs an analysis on your system to determine which (if any) patches will be removed by upgrading to a Oracle Solaris update.
pool	A logical group of devices describing the layout and physical characteristics of the available ZFS storage. Space for datasets is allocated from a pool.
primary boot archive	A boot archive that is used to boot the Oracle Solaris OS on a system. This boot archive is sometimes called the primary boot archive. See <i>boot archive</i> .
profile	A text file that defines how to install the Oracle Solaris software when using the custom JumpStart method. For example, a profile defines which software group to install. Every rule specifies a profile that defines how a system is to be installed when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule. See also <i>rules file</i> .
RAID-0 volume	A class of volume that can be a stripe or a concatenation. These components are also called submirrors. A stripe or concatenation is the basic building block for mirrors.
RAID-1 volume	A class of volume that replicates data by maintaining multiple copies. A RAID-1 volume is composed of one or more RAID-0 volumes called <i>submirrors</i> . A RAID-1 volume is sometimes called a <i>mirror</i> .
RAID-Z storage pool	A virtual device that stores data and parity on multiple disks that can be used as a ZFS storage pool. RAID-Z is similar to RAID-5.
Reduced Network Support Software Group	A software group that contains the minimum code that is required to boot and run a Oracle Solaris system with limited network service support. The Reduced Networking Software Group provides a multiuser text-based console and system administration utilities. This software group also enables the system to recognize network interfaces, but does not activate network services.
root	The top level of a hierarchy of items. Root is the one item from which all other items are descended. See <i>root directory</i> or <i>root (/) file system</i> .

root (/) file system	The top-level file system from which all other file systems stem. The root (/) file system is the base on which all other file systems are mounted, and is never unmounted. The root (/) file system 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.
root directory	The top-level directory from which all other directories stem.
rule	A series of values that assigns one or more system attributes to a profile. A rule is used in a custom JumpStart installation.
rules file	A text file that contains a rule for each group of systems or single systems that you want to install automatically. Each rule distinguishes a group of systems, based on one or more system attributes. The rules file links each group to a profile, which is a text file that defines how the Oracle Solaris software is to be installed on each system in the group. A rules file is used in a custom JumpStart installation. See also <i>profile</i> .
rules.ok file	A generated version of the rules file. The rules.ok file is required by the custom JumpStart installation software to match a system to a profile. You <i>must</i> use the check script to create the rules.ok file.
slice	The unit into which the disk space is divided by the software.
software group	A logical grouping of the Oracle Solaris software (clusters and packages). During a Oracle Solaris installation, you can install one of the following software groups: Core, End User Oracle Solaris Software, Developer Oracle Solaris Software, or Entire Oracle Solaris Software, and for SPARC systems only, Entire Oracle Solaris Software Group Plus OEM Support.
state database	A database that stores information about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a <i>state database replica</i> . The state database tracks the location and status of all known state database replicas.
state database replica	A copy of a state database. The replica ensures that the data in the database is valid.
submirror	See <i>RAID-0 volume</i> .
swap space	A slice or file that temporarily holds the contents of a memory area till it can be reloaded in memory. Also called the /swap or swap volume.
unmount	The process of removing access to a directory on a disk that is attached to a machine or to a remote disk on a network.
update	An installation, or to perform an installation, on a system that changes software that is of the same type. Unlike an upgrade, an update might downgrade the system. Unlike an initial installation, software of the same type that is being installed must be present before an update can occur.
upgrade	<p>An installation that merges files with existing files and preserves modifications where possible.</p> <p>An upgrade of the Oracle Solaris OS merges the new version of the Oracle Solaris OS with the existing files on the system's disk or disks. An upgrade saves as many modifications as possible that you have made to the previous version of the Oracle Solaris OS.</p>

upgrade option	An option that is presented by the Oracle Solaris installation. The upgrade procedure merges the new version of Oracle Solaris with existing files on your disk or disks. An upgrade also saves as many local modifications as possible since the last time Oracle Solaris was installed.
/usr file system	A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large /usr file system with a server rather than maintaining a local copy minimizes the overall disk space that is required to install and run the Oracle Solaris software on a system.
/var file system	A file system or directory (on standalone systems) that contains system files that are likely to change or grow over the life of the system. These files include system logs, vi files, mail files, and UUCP files.
virtual device	A logical device in a ZFS pool, which can be a physical device, a file, or a collection of devices.
volume	<p>A group of physical slices or other volumes that appear to the system as a single logical device. A volume is functionally identical to a physical disk for the purposes of an application or file system.</p> <p>In some command-line utilities, a volume is called a metadvice. Volume is also called <i>pseudo device</i> or <i>virtual device</i> in standard UNIX terms.</p>
Volume Manager	A program that provides a mechanism to administer and obtain access to the data on DVD-ROMs, CD-ROMs, and diskettes.
ZFS	A file system using storage pools to manage physical storage.
zone	See <i>non-global zone</i>

Index

A

- activating a boot environment
 - description, 38
 - failure, description, 39
 - synchronizing files, description, 54
 - tasks, 104
- archives
 - creating an empty boot environment for, 76
 - example of installing, 35
 - installing on a boot environment, 100

B

- boot: cannot open /kernel/unix message, 212
- boot environment, failure of, description, 39
- bootparams file, updating, 217

C

- Can't boot from file/device message, 212
- cancel a Live Upgrade job, 124
- CHANGE DEFAULT BOOT DEVICE message, 218
- change name of a boot environment, 126
- CLIENT MAC ADDR error message, 217
- clock gained xxx days message, 212
- commands for Live Upgrade, 209
- compare boot environments, 124
- compare file systems, non-global zones, 145
- concatenation, description, 29
- copying, file systems, 123

creating

- a boot environment, description, 22
- a boot environment, tasks, 64, 67, 68
- RAID-1 volume (mirror), description, 27
- task map, 59–60
- tasks
 - for ZFS, 181
- critical file systems, definition, 22
- customizing content, 53

D

- deleting, a boot environment, 125
- disk space requirements, 46
- displaying, name of a boot environment, 126

E

- /etc/bootparams file, enabling JumpStart directory
 - access, 217
- examples, 147
 - complete process
 - complete process, 147
 - creating mirrors, 82
 - creating RAID-1 volumes, 81
 - upgrading a RAID-1 volume, 156, 159

F

- failed upgrade
 - rebooting problems, 222
 - recovery, 111
- files and file systems
 - creating RAID-1 volumes (mirrors), description, 27
 - description, 22
 - estimating size, 46
 - guidelines for creating, 48
 - guidelines for selecting a slice, 49
 - sharing file systems between boot environments, 52

I

- installing
 - a Flash Archive, 100
 - a Flash Archive with a profile, 103
 - packages, 60

K

- keywords
 - profile, 94
 - volumes, 79

L

- le0: No carrier - transceiver cable problem
 - message, 212
- lumount command, non-global zones, 146

M

- migrating from UFS to ZFS, 181
 - with non-global zones, 199
- mirror, *See* RAID-1 volume

N

- No carrier - transceiver cable problem message, 212

non-global zones

- administering boot environments, 144
- compare file systems, 145
- graphic of, 132
- list file systems, 144
- lumount command, 146
- migrating from UFS to ZFS, 199
- overview, 131
- separate file system, 137
- step-by-step procedure, 137
- upgrade example, 143
- Not a UFS filesystem message, 212

O

- overview, 19
 - for ZFS, 165
 - graphic, 20

P

- packages
 - adding, 47, 87
 - requirements for Live Upgrade, 229
 - requirements when using custom JumpStart, 229
- Patch Analyzer, 239–240
- patches
 - adding, 47, 87
 - checking patch levels, 44, 62
- planning, 43
 - for ZFS, 177
- profile keywords
 - forced_deployment
 - description and values, 99
 - local_customization
 - description and values, 99
- profiles
 - example, 94
 - example for differential archives, 95

R

RAID-0 volume, description, 29
 RAID-1 volume (mirror)
 description, 27, 29
 example of creating, 81, 82
 example of creating and upgrading, 156
 example of migrating to Solaris Volume Manager
 volumes, 159
 requirements, 50
 required packages, 45
 requirements, to use Live Upgrade, 43
 root (/) file systems, package requirements for an
 inactive boot environment, 229
 RPC Timed out message, 216

S

selecting slices for RAID-1 volumes (mirrors), 50
 shareable file systems, definition, 22
 slices, guidelines for selecting, 49
 Solaris Volume Manager
 commands used with Live Upgrade, 51
 example
 detaching and upgrading a RAID-1 volume, 156
 migrating to a RAID-1 volume, 159
 state database, description, 29
 status, display boot environment, 122
 submirror, description, 29
 swap file systems, guidelines for selecting a slice, 52

T

testing, profiles, 96
 timed out RPC error, 216
 token ring card, booting error with, 216
 transceiver cable problem message, 212
 troubleshooting
 booting from network with DHCP, 217
 booting from wrong server, 217
 general installation problems
 booting from the network with DHCP, 217
 booting the system, 217

U

Unknown client error message, 211
 upgrade
 a boot environment
 a boot environment, 86
 description, 33
 examples, 147, 156, 159
 failed upgrade, 222
 failed upgrade recovery, 111
 guidelines for, 86
 non-global zones
 administering boot environments, 144
 compare boot environments, 145
 example, 143
 graphic of, 132
 list file systems, 144
 lumount command, 146
 overview, 131
 separate file system, 137
 step-by-step procedure, 137
 task map, 85–86
 tasks, 86
 install a Flash Archive, 100
 to a Oracle Solaris Update release, 239–240

V

view configuration of boot environments, non-global
 zones, 144
 viewing, configuration of boot environments, 128
 volume
 RAID-0, description, 29
 RAID-1, description, 29
 Volume Manager, *See* Solaris Volume Manager

W

WARNING: CHANGE DEFAULT BOOT
 DEVICE, 218
 WARNING: clock gained xxx days message, 212

Z

ZFS

- creating a boot environment, 181
 - from another source, 197
 - in a new pool, 192
 - in the same pool, 188
- migrating
 - with non-global zones, 199
- migrating from UFS to ZFS, 181
- overview, 165
- planning, 177