



Solaris 9 Installation Guide

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

Part No: 816-5102-10
September 2002

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

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

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

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

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

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

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

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés

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

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

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

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

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



020605@3984



Contents

Preface	15
1 Planning for Solaris Installation or Upgrade (Topics)	19
2 Planning for a Solaris Installation or Upgrade (Overview)	21
Task Map: Installing or Upgrading the Solaris Software	21
Initial Installation or Upgrade?	22
Initial Installation	23
Upgrade	23
System Requirements	24
Memory Requirement	24
Requirements When Using the Solaris 9 Installation CD	24
Installing From the Network or From DVD or CDs?	25
Using DVD Media	25
3 Choosing a Solaris Installation Method (Overview)	27
Solaris Web Start Installation Program	27
Solaris <code>suninstall</code> Program	28
Custom JumpStart Installation Method	28
Web Start Flash Installation Feature	29
Solaris Live Upgrade Method	30
Factory JumpStart Installation Method	30

4	Preparing for Solaris Installation or Upgrade (Topics)	31
5	Guidelines for Allocating Disk Space (Planning)	33
	General Disk Space Planning and Recommendations	33
	Disk Space Recommendations for Software Groups	34
6	Gathering Information Before Installation or Upgrade (Planning)	37
	Checklist for Installation	37
	Worksheet for Installation	38
	Checklist for Upgrading	41
	Worksheet for Upgrading	42
7	Preconfiguring System Configuration Information (Tasks)	47
	Advantages of Preconfiguring System Configuration Information	47
	Ways to Preconfigure System Configuration Information	48
	Preconfiguring With the <code>sysidcfg</code> File	49
	Syntax Rules for the <code>sysidcfg</code> File	50
	<code>sysidcfg</code> File Keywords	50
	▼ To Create a <code>sysidcfg</code> Configuration File	53
	Preconfiguring With the Name Service	54
	▼ To Preconfigure the Locale Using NIS	54
	▼ To Preconfigure the Locale Using NIS+	56
	Preconfiguring Power Management Information	57
8	Upgrading the Solaris Operating Environment (Planning)	59
	Upgrading	59
	Using Solaris Live Upgrade	60
	Using Custom JumpStart to Upgrade	60
	Upgrading With Disk Space Reallocation	61
	Backing Up Systems Before Upgrading	62
9	Preparing to Install From the Network (Topics)	63
10	Preparing to Install From the Network (Overview)	65
	Planning for a Network Installation Introduction	65

	Required Servers for Network Installation	65
	Using DHCP to Provide Network Installation Parameters	67
11	Preparing to Install From the Network With DVD Media (Tasks)	69
	Task Map: Preparing to Install From the Network With DVD Media	69
	Creating an Install Server With DVD Media	70
	▼ To Create an Install Server With DVD Media	70
	Creating a Boot Server on a Subnet With a DVD Image	73
	▼ To Create a Boot Server on a Subnet With a DVD Image	74
	Adding Systems to Be Installed From the Network	75
	▼ To Add Systems to Be Installed From the Network With add_install_client From a Server	76
12	Preparing to Install From the Network With CD Media (Tasks)	79
	Task Map: Preparing to Install From the Network With CD Media	80
	Creating an Install Server With CD Media	80
	▼ To Create an Install Server With CD Media	81
	Creating a Boot Server on a Subnet With a CD Image	85
	▼ To Create a Boot Server on a Subnet With a CD Image	85
	Adding Systems to Be Installed From the Network	87
	▼ To Add Systems to Be Installed From the Network With add_install_client From a Server	88
13	Preparing to Install From the Network (Command Reference)	91
14	Using the Solaris Web Start Program (Tasks)	93
	Solaris Web Start Program GUI or CLI	93
	Performing an Installation or Upgrade With the Solaris Web Start Program	94
	Task Map: Performing a Solaris Web Start Installation	94
	▼ To Perform an Installation or Upgrade With the Solaris Web Start Program	95
	Solaris Web Start Post-Installation and Upgrade Tasks	98
	▼ To Correct Local Modifications After Upgrading	98
15	Using the Solaris suninstall Program (Tasks)	101
	Solaris suninstall Program	101

Performing an Installation or Upgrade With the Solaris <code>suninstall</code> Program	101
Task Map: Performing a Solaris <code>suninstall</code> Program Installation	102
▼ To Perform an Installation or Upgrade With the Solaris <code>suninstall</code> Program	102
Solaris <code>suninstall</code> Program Post-Upgrade Task	104
▼ To Correct Local Modifications After Upgrading	104
16 Web Start Flash Installation Feature (Topics)	105
17 Web Start Flash (Overview and Planning)	107
Web Start Flash Introduction	107
Planning Your Web Start Flash Installation	108
Designing the Installation of the Master System	108
Planning the Creation of a Web Start Flash Archive	110
Planning the Installation of Web Start Flash Archives	111
18 Creating Web Start Flash Archives (Tasks)	113
Task Map: Creating Web Start Flash Archives	113
Creating Web Start Flash Archives Tasks	113
▼ To Install the Master System	114
▼ To Create a Web Start Flash Archive	114
19 Installing Web Start Flash Archives (Tasks)	117
Installing Web Start Flash Archives With Solaris Live Upgrade	117
Installing Web Start Flash Archives With the Solaris Web Start Program	117
▼ To Install a Web Start Flash Archive With the Solaris Web Start Program	118
Installing Web Start Flash Archives With the Solaris <code>suninstall</code> Program	119
▼ To Install a Web Start Flash Archive With the Solaris <code>suninstall</code> Program	119
Installing Web Start Flash Archives With a Custom JumpStart Installation	120
▼ To Install a Web Start Flash Archive With a Custom JumpStart Installation	120
Web Start Flash Custom JumpStart Profile Examples	121
20 Web Start Flash (Reference)	123
Web Start Flash Archive Sections	123

	Web Start Flash Keywords	124
	General Keywords	124
	Identification Section Keywords	124
	Web Start Flash Commands	126
	flarcreate	126
	flar	128
21	Custom JumpStart Installation (Topics)	131
22	Custom JumpStart (Overview)	133
	Custom JumpStart Introduction	133
	How the JumpStart Program Installs Solaris Software	134
23	Preparing Custom JumpStart Installations (Tasks)	139
	Task Map: Preparing Custom JumpStart Installations	139
	Creating a Profile Server for Networked Systems	141
	▼ To Create a JumpStart Directory on a Server	141
	▼ To Allow All Systems Access to the Profile Server	143
	Creating a Profile Diskette for Standalone Systems	144
	▼ To Create a Profile Diskette	145
	Creating the rules File	147
	Syntax of the rules File	147
	▼ To Create a rules File	148
	rules File Example	149
	Creating a Profile	150
	Syntax of Profiles	150
	▼ To Create a Profile	151
	Profile Examples	151
	Testing a Profile	153
	▼ To Create a Temporary Solaris 9 Environment to Test a Profile	154
	▼ To Test a Profile	155
	Profile Test Examples	156
	Validating the rules File	157
	▼ To Validate the rules File	157

24	Using Optional Custom JumpStart Features (Tasks)	159
	Creating Begin Scripts	159
	Important Information About Begin Scripts	159
	Creating Derived Profiles With a Begin Script	160
	Creating Finish Scripts	161
	Important Information About Finish Scripts	161
	▼ To Add Files With a Finish Script	161
	Adding Packages or Patches With a Finish Script	162
	Customizing the Root Environment With a Finish Script	163
	Setting a System's Root Password With a Finish Script	164
	Installing Software With Web Start Installation Programs With Finish Scripts	165
	Creating a Compressed Configuration File	166
	▼ To Create a Compressed Configuration File	166
	Compressed Configuration File Example	167
	Creating Disk Configuration Files	167
	▼ To Create a Disk Configuration File	167
	Disk Configuration File Example	168
	Using a Site-Specific Installation Program	169
25	Creating Custom Rule and Probe Keywords (Tasks)	171
	Probe Keywords	171
	Creating a custom_probes File	172
	Syntax of the custom_probes File	172
	Syntax of Function Names in custom_probes	173
	▼ To Create a custom_probes File	173
	Examples of a custom_probes File and Keyword	173
	Validating the custom_probes File	175
	▼ To Validate the custom_probes File	175
26	Performing a Custom JumpStart Installation (Tasks)	177
	Performing a Custom JumpStart Installation	177
	Task Map: Setting Up a System for a Custom JumpStart Installation	177
	▼ To Perform an Installation or Upgrade With the Custom JumpStart Program	178

27	Installing With Custom JumpStart (Examples)	181
	Sample Site Setup	181
	Create an Install Server	183
	Create a Boot Server for Marketing Systems	184
	Create a JumpStart Directory	184
	Share the JumpStart Directory	185
	Create the Engineering Group's Profile	185
	Create the Marketing Group's Profile	185
	Update the rules File	186
	Validate the rules File	187
	Set Up Engineering Systems to Install From the Network	187
	Set Up Marketing Systems to Install From the Network	188
	Boot the Systems and Install Solaris 9 Software	188
28	Custom JumpStart (Reference)	189
	Rule Keywords and Values	189
	Profile Keywords and Values	194
	Profile Keywords at a Glance	194
	Profile Keyword Descriptions and Examples	196
	Custom JumpStart Environment Variables	217
	Probe Keywords and Values	220
29	Solaris Live Upgrade (Topics)	223
30	Solaris Live Upgrade (Overview)	225
	Solaris Live Upgrade Introduction	225
	Solaris Live Upgrade Process	226
31	Solaris Live Upgrade (Planning)	239
	Solaris Live Upgrade System Requirements	239
	Solaris Live Upgrade Disk Space Requirements	240
	Managing Packages and Patches With Solaris Live Upgrade	240
	Required Packages	241
	Checking System Patch Levels	242
	Guidelines for Creating File Systems With the <code>lucreate</code> Command	242
	Guidelines for Selecting Slices for File Systems	243

	Guidelines for Selecting a Slice for the root (/) File System	243
	Guidelines for Selecting Slices for root (/) Mirrors and Metadevices	244
	Guidelines for Selecting a Slice for a swap File System	245
	Guidelines for Selecting Slices for Shareable File Systems	245
	Using Solaris Live Upgrade From a Remote System	246
32	Using Solaris Live Upgrade to Create a Boot Environment (Tasks)	247
	About Solaris Live Upgrade Interfaces	247
	Using Solaris Live Upgrade Menus	248
	Task Map: Installing Solaris Live Upgrade and Creating Boot Environments	249
	Installing Solaris Live Upgrade	249
	▼ To Install Solaris Live Upgrade	249
	Starting and Stopping Solaris Live Upgrade (Character Interface)	250
	▼ To Start Solaris Live Upgrade Menus	250
	▼ To Stop Solaris Live Upgrade Menus	251
	Creating a New Boot Environment	251
	▼ To Create a Boot Environment (Character Interface)	252
	▼ To Create a Boot Environment for the First Time (Command-Line Interface)	257
	▼ To Create a Boot Environment and Merge File Systems (Command-Line Interface)	259
	▼ To Create a Boot Environment and Split File Systems (Command-Line Interface)	260
	▼ To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)	262
	▼ To Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)	264
	▼ To Create a Boot Environment and Copy a Shareable File System (Command-Line Interface)	266
	▼ To Create a Boot Environment From a Different Source (Command-Line Interface)	267
	▼ To Create a Boot Environment for a Flash Archive (Command-Line Interface)	268
33	Upgrading With Solaris Live Upgrade (Tasks)	271
	Task Map: Upgrading a Boot Environment	272
	Upgrading a Boot Environment	272
	Upgrading Metadevices and Volumes	273
	Managing Packages and Patches With Solaris Live Upgrade	274

	▼ To Upgrade an Operating System Image on a Boot Environment (Character Interface)	275
	▼ To Upgrade an Operating System Image on a Boot Environment (Command-Line Interface)	276
	▼ To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)	276
	Installing Web Start Flash Archives on a Boot Environment	278
	Upgrading Metadevices and Volumes	278
	▼ To Install Web Start Flash Archives on a Boot Environment (Character Interface)	280
	▼ To Install Web Start Flash Archives on a Boot Environment (Command-Line Interface)	281
	Activating a Boot Environment	282
	▼ To Activate a Boot Environment (Character Interface)	283
	▼ To Activate a Boot Environment (Command-Line Interface)	284
	▼ To Activate a Boot Environment and Synchronize Files (Command-Line Interface)	285
	Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)	286
	▼ To Fall Back Despite Successful New Boot Environment Creation	286
	▼ To Fall Back From a Failed Boot Environment Activation	287
	▼ To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image	288
34	Maintaining Solaris Live Upgrade Boot Environments (Tasks)	291
	Overview of Solaris Live Upgrade Maintenance	292
	Displaying the Status of All Boot Environments	293
	▼ To Display the Status of All Boot Environments (Character Interface)	293
	▼ To Display the Status of All Boot Environments (Command-Line Interface)	294
	Updating a Previously Configured Boot Environment	294
	▼ To Update a Previously Configured Boot Environment (Character Interface)	295
	▼ To Update a Previously Configured Boot Environment (Command-Line Interface)	295
	Canceling a Scheduled Create, Upgrade, or Copy Job	296
	▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Character Interface)	296
	▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Command-Line Interface)	297

Comparing Boot Environments	297
▼ To Compare Boot Environments (Character Interface)	297
▼ To Compare Boot Environments (Command-Line Interface)	298
Deleting an Inactive Boot Environment	299
▼ To Delete an Inactive Boot Environment (Character Interface)	299
▼ To Delete an Inactive Boot Environment (Command-Line Interface)	299
Displaying the Name of the Active Boot Environment	300
▼ To Display the Name of the Active Boot Environment (Character Interface)	300
▼ To Display the Name of the Active Boot Environment (Command-Line Interface)	300
Changing the Name of a Boot Environment	301
▼ To Change the Name of an Inactive Boot Environment (Character Interface)	301
▼ To Change the Name of an Inactive Boot Environment (Command-Line Interface)	301
Adding or Changing a Description Associated With a Boot Environment Name	302
▼ To Add or Change a Description for a Boot Environment Name With Text	302
▼ To Add or Change a Description for a Boot Environment Name With a File	303
▼ To Determine a Boot Environment Name From a Text Description	303
▼ To Determine a Boot Environment Name From a Description in a File	304
▼ To Determine a Boot Environment Description From a Name	304
Viewing the Configuration of a Boot Environment	305
▼ To View the Configuration of Each Inactive Boot Environment (Character Interface)	305
▼ To View the Configuration of a Boot Environment (Command-Line Interface)	305
35 Solaris Live Upgrade (Examples)	307
Example of Upgrading With Solaris Live Upgrade (Command-Line Interface)	307
Install Live Upgrade on the Active Boot Environment	307
Create a Boot Environment	308
Upgrade the Inactive Boot Environment	308
Check If Boot Environment Is Bootable	308
Activate the Inactive Boot Environment	309
Fall Back to the Source Boot Environment	309
Example of Upgrading Using Solaris Live Upgrade (Character Interface)	310
Install Live Upgrade on the Active Boot Environment	310

	Create a Boot Environment	311
	Upgrade the Inactive Boot Environment	312
	Activate the Inactive Boot Environment	313
36	Solaris Live Upgrade (Command Reference)	315
37	Solaris Software Reference (Topics)	317
38	Organization of Solaris 9 Media (Reference)	319
	Solaris 9 Media	319
	Directory Organization of Solaris 9 Media	320
	Solaris 9 DVD Directory Structure	321
	Solaris 9 Installation CD Directory Structure	322
	Solaris 9 Software CDs Directory Structures	322
	Solaris 9 Languages CD Directory Structure	324
39	Platform Names and Groups (Reference)	327
40	Locale Values (Reference)	329
A	Troubleshooting (Tasks)	337
	Problems Setting Up Network Installations	337
	Problems With Booting a System	338
	Booting From Media, Error Messages	338
	Booting From Media, General Problems	339
	Booting From the Network, Error Messages	339
	Booting From the Network, General Problems	341
	Initial Installation of the Solaris 9 Operating Environment	342
	Upgrading the Solaris 9 Operating Environment	342
	Upgrading, Error Messages	342
	Upgrading, General Problems	343
	▼ To Continue Upgrading After a Failed Upgrade	344
	System Panics When Upgrading With Solaris Live Upgrade Running Veritas VxVm	344

B	Installing or Upgrading Remotely (Tasks)	347
	Using the Solaris Web Start Program to Install or Upgrade From a Remote DVD-ROM or CD-ROM	347
	▼ To Install or Upgrade From a Remote DVD-ROM and CD-ROM	348
C	Additional SvR4 Packaging Requirements (Reference)	351
	Packaging Requirements Overview	351
	Custom JumpStart Program and Solaris Live Upgrade Alternate Root (/) Requirements	352
	Differences Between \$PKG_INSTALL_ROOT and \$BASEDIR Overview	353
	Solaris Live Upgrade Alternate Boot Environment Compliance	354
	Upgrade Compliance With the Custom JumpStart Program	356
D	Upgrading to a Solaris Update Release (Tasks)	359
	Upgrading to a Solaris Update Release	359
	▼ To Run the analyze_patches Script	360
	▼ To Review the Patch Analyzer Output	360
E	Solaris 9 Updates	363
	Solaris 9 9/02 release.	363
	Glossary	365
	Index	375

Preface

The *Solaris 9 Installation Guide* describes how to install and upgrade the Solaris™ 9 operating environment on both networked and non-networked SPARC™ systems.

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

Who Should Use This Book

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

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

Related Books

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

TABLE P-1 Related Information

Information	Description
<i>System Administration Guide: Basic Administration</i>	Describes how to back up system files
<i>Solaris 9 (SPARC Platform Edition) Online Release Notes</i>	Describes any bugs, known problems, software that is being discontinued, and patches that are related to the Solaris release
<i>Solaris 9 Sun Hardware Platform Guide</i>	Contains information about supported hardware
<i>Solaris 9 Package List</i>	Lists and describes the packages in the Solaris 9 operating environment

Accessing Sun Documentation Online

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

Typographic Conventions

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

TABLE P-2 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with on-screen computer output	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename.</code>

TABLE P-2 Typographic Conventions (Continued)

Typeface or Symbol	Meaning	Example
<i>AaBbCc123</i>	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You must be <i>root</i> to do this.

Shell Prompts in Command Examples

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

TABLE P-3 Shell Prompts

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

Planning for Solaris Installation or Upgrade (Topics)

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

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

Planning for a Solaris Installation or Upgrade (Overview)

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

- “Task Map: Installing or Upgrading the Solaris Software” on page 21
- “Initial Installation or Upgrade?” on page 22
- “System Requirements” on page 24
- “Installing From the Network or From DVD or CDs?” on page 25
- “Using DVD Media” on page 25

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

Task Map: Installing or Upgrading the Solaris Software

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

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

Task	Description	For Instructions
Choose initial installation or upgrade.	Decide if you want to perform an initial installation or an upgrade.	“Initial Installation or Upgrade?” on page 22

TABLE 2-1 Task Map: Installing or Upgrading the Solaris Software (Continued)

Task	Description	For Instructions
Review system requirements.	Determine if your system meets the minimum requirements to install or upgrade.	"System Requirements" on page 24
Choose an installation method.	The Solaris operating environment provides several methods for installation or upgrade. Choose the installation method that is most appropriate for your environment.	Chapter 3
Plan and allocate disk space.	Allocate disk space on your system for the components of the Solaris operating environment that you want to install.	Chapter 5
Choose an installation location.	You can install the Solaris software from local media or from the network. Decide on an installation location that is most appropriate for your environment.	"Installing From the Network or From DVD or CDs?" on page 25
Gather information about your system.	Use the checklist and complete the worksheet to collect all of the information that you need to install or upgrade.	Chapter 6
(Optional) Preconfigure system information.	You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.	Chapter 7
(Optional) Prepare to install the Solaris software from the network.	If you chose to install the Solaris software from the network, create an install server, create a boot server (if necessary), and set up the systems to be installed from the network.	Chapter 12
(Upgrade only) Perform the pre-upgrade tasks.	Back up your system, determine if you can upgrade with disk space reallocation, and search for patches that a Solaris Update release might override.	Chapter 8
Install or upgrade.	Use the Solaris installation method that you chose to install or upgrade the Solaris software.	The chapter or chapters that provide detailed instructions for the installation method

Initial Installation or Upgrade?

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

Initial Installation

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

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

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

Upgrade

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

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

```
$ uname -a
```

You can use any of the installation methods to upgrade the Solaris operating environment. For limitations on upgrading with Solaris Live Upgrade, see "Solaris Live Upgrade System Requirements" on page 239.

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

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

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

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

You can use the Patch Analyzer to determine which patches, if any, will be removed by upgrading to the Solaris 9 Update release. For detailed instructions about using the Patch Analyzer, refer to “Upgrading to a Solaris Update Release” on page 359.

System Requirements

Verify that your system meets the requirements to install or upgrade to the Solaris operating environment.

Memory Requirement

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

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

Requirements When Using the Solaris 9 Installation CD

When you use the Solaris 9 Installation CD and the Solaris Web Start program to upgrade, you must have a slice on the disk that does not store files. The swap slice is preferred, but you can use any slice that is not located in any of the “upgradable” root slices that are listed in `/etc/vfstab`. The size of this slice must be at least 512 Mbytes.

Installing From the Network or From DVD or CDs?

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

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

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

Installing the Solaris software from the network requires initial setup. For detailed instructions on preparing to install from the network, refer to Chapter 12.

Using DVD Media

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

```
ok boot cdrom
```


Choosing a Solaris Installation Method (Overview)

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

- “Solaris Web Start Installation Program” on page 27
- “Solaris `suninstall` Program” on page 28
- “Custom JumpStart Installation Method” on page 28
- “Web Start Flash Installation Feature” on page 29
- “Solaris Live Upgrade Method” on page 30
- “Factory JumpStart Installation Method” on page 30

Solaris Web Start Installation Program

The Solaris Web Start installation program on the Solaris 9 DVD or Solaris 9 Installation CD can be run with a graphical user interface (GUI) or with a command-line interface (CLI). The Solaris Web Start program guides you step-by-step through installing or upgrading the Solaris software and additional software. You can install with the default option, or you can use the customize option to install only the software you want.

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

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

For detailed instructions, refer to Chapter 14.

Solaris `suninstall` Program

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

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

For detailed instructions, refer to Chapter 15.

Custom JumpStart Installation Method

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

If you know the Solaris operating environment and the shell, and have multiple systems to install, the custom JumpStart installation method might be the most efficient way for you to install your systems.

If you plan to install only a few systems, this installation method is less efficient. The creation of a custom JumpStart environment might be so time-consuming that you do not save time by using this installation method.

For detailed instructions, refer to Chapter 23.

Web Start Flash Installation Feature

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

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

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

If you have many different configurations that you want to install on your systems, you need a Web Start Flash archive for each system. Web Start Flash archives are large files and require a significant amount of disk space. Also, after you create a Web Start Flash archive, you cannot change the archive. If you have many different installation configurations or if you want the flexibility to change your installation configuration, you might consider using the custom JumpStart installation method.

For detailed instructions, refer to Chapter 17.

Solaris Live Upgrade Method

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

For a detailed overview and instructions, refer to Chapter 29.

Factory JumpStart Installation Method

The JumpStart installation method automatically installs the Solaris software on a new SPARC system when you insert the Solaris 9 DVD or Solaris 9 Software 1 of 2 CD into the drive and turn on the system. A default profile is selected that is based on the model and disk size of the system. The profile determines which software components are installed on the system. You are not prompted for system configuration information and you cannot choose which software is installed.

A JumpStart boot image, which is required to use this installation method, is preinstalled on all new SPARC based systems. If you have an older SPARC based system, you can add the JumpStart installation method to the system by using the `re-preinstall(1M)` command.

Preparing for Solaris Installation or Upgrade (Topics)

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

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

Guidelines for Allocating Disk Space (Planning)

This chapter describes general guidelines for planning the disk space that you need to install or upgrade the Solaris operating environment. This chapter contains the following sections:

- “General Disk Space Planning and Recommendations” on page 33
- “Disk Space Recommendations for Software Groups” on page 34

General Disk Space Planning and Recommendations

Before you install the Solaris 9 software, you can determine if your system has enough disk space by doing some high-level planning. If you plan your disk space needs before you install, you can add more disks to your system, if you need them, before you install the Solaris 9 software. If you are upgrading, review “Upgrading With Disk Space Reallocation” on page 61.

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

- Allocate additional disk space for each language you select to install, for example, Chinese, Japanese, Korean.
- If you intend to support printing or mail, allocate additional disk space in the `/var` file system.
- If you intend to use the crash dump feature `savecore(1M)`, allocate double the amount of your physical memory in the `/var` file system.
- If a server provides home directory file systems for users on other systems, allocate additional disk space on the server. By default, home directories are usually located in the `/export` file system.

- Allocate at least 512 Mbytes of swap space.
- Allocate space for the Solaris software group you want to install. The recommended disk space for the software groups is in Table 5–1. When you are planning disk space, remember that you can add or remove individual software packages from the software group that you select.
- Create a minimum number of file systems. By default, the Solaris installation methods create only root (/) and /swap. When space is allocated for OS services, the /export directory is also created. For each file system that you create, allocate an additional 30 percent more disk space than you need to enable you to upgrade to future Solaris versions. Each new Solaris release needs approximately 10 percent more disk space than the previous release. By allocating an additional 30 percent of disk space for each file system, you allow for several Solaris upgrades before you need to reslice your system.
- Allocate additional disk space for additional software or third-party software.

If you are using Solaris Live Upgrade, see “Solaris Live Upgrade Disk Space Requirements” on page 240.

Disk Space Recommendations for Software Groups

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

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

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

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

Note – Swap space is included in the disk space recommendations.

TABLE 5-1 Disk Space Recommendations for Software Groups

Software Group	Recommended Disk Space
Entire Solaris Software Group Plus OEM Support	2.9 Gbytes
Entire Solaris Software Group	2.7 Gbytes
Developer Solaris Software Group	2.4 Gbytes
End User Solaris Software Group	2.0 Gbytes

Gathering Information Before Installation or Upgrade (Planning)

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

- “Checklist for Installation” on page 37
- “Worksheet for Installation” on page 38
- “Checklist for Upgrading” on page 41
- “Worksheet for Upgrading” on page 42

Checklist for Installation

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

- If you intend to install the Solaris software on a system through a `tip(1)` line, ensure that your window display is at least 80 columns wide and 24 rows long.
To determine the current dimensions of your `tip` window, use the `stty` command. For more information, see the man page, `stty(1)`.
- If the system is part of a network, verify that an Ethernet connector or similar network adapter is plugged into your system.
- If you are installing with the Solaris Web Start program by using the Solaris 9 Installation CD, review “Requirements When Using the Solaris 9 Installation CD” on page 24.
- Verify that you have enough disk space. For more information, refer to Chapter 5.
- When you are using DVD media and are asked to boot from the `ok` prompt, always type the following command: `boot cdrom`.
- Review the *Solaris 9 Release Notes* and vendor release notes to ensure that the software you use is supported in the new Solaris release.
- Review the *Solaris 9 Sun Hardware Platform Guide* to ensure that your hardware is supported.

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

Worksheet for Installation

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

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

TABLE 6-1 Installation Worksheet

Info Needed to Install	Description/Example	Enter Your Answers Here:
Network	Is the system connected to a network?	Yes/No
DHCP	Can the system use Dynamic Host Configuration Protocol (DHCP) to configure its network interfaces?	Yes/No
Host Name	Host name that you choose for the system.	
IP Address	If you are not using DHCP, supply the IP address for the system. Example: 129.200.9.1	
Subnet	If you are not using DHCP, is the system part of a subnet? If yes, what is the netmask of the subnet? Example: 255.255.0.0	Yes/No
IPv6	Do you want to enable IPv6 on this machine?	Yes/No

TABLE 6-1 Installation Worksheet (Continued)

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

TABLE 6-1 Installation Worksheet (Continued)

Info Needed to Install	Description/Example	Enter Your Answers Here:
Default Router	<p>Do you want to specify a default IP router (gateway) or let the Solaris Web Start installation program find one?</p> <p>If you want to specify a default router, provide the following information.</p> <p style="text-align: right;">Router IP address:</p>	Specify One/Find One
Time Zone	How do you want to specify your default time zone?	Geographic region Offset from GMT Time zone file
Locales	For which geographic regions do you want to install support?	
Power Management	Do you want to use Power Management?	Yes/No
Proxy Server Configuration (only available in the Solaris Web Start program)	<p>Do you have a direct connection to the Internet or do you need to use a proxy server to gain access to the Internet?</p> <p>If you use a proxy server, provide the following information.</p> <p style="text-align: right;">Host: Port:</p>	Direct Connection/Proxy Server
Automatic reboot or CD/DVD ejection	Reboot automatically after software installation? Eject CD/DVD automatically after software installation?	Yes/No Yes/No
Software Group	Which Solaris Software Group do you want to install?	Entire Plus OEM Entire Developer End User Core
Custom Package Selection	<p>Do you want to add or remove software packages from the Solaris Software Group that you install?</p> <p>Note – When you select which packages to add or remove, you need to know about software dependencies and how Solaris software is packaged.</p>	

TABLE 6-1 Installation Worksheet (Continued)

Info Needed to Install	Description/Example	Enter Your Answers Here:
64-bit	Do you want to install support for 64-bit applications?	Yes/No
Select Disks	On which disks do you want to install the Solaris software? Example: c0t0d0	
Preserve Data	Do you want to preserve any data that exists on the disks where you are installing the Solaris software?	Yes/No
Auto-layout File Systems	Do you want the installation program to automatically lay out file systems on your disks? If yes, which file systems should be used for auto-layout? Example: /, /opt, /var If no, you must provide file system configuration information.	Yes/No
Mount Remote File Systems (only available in the Solaris <code>suninstall</code> program)	Does this system need to access software on another file system? If yes, provide the following information about the remote file system. Server: IP Address: Remote File System: Local Mount Point:	Yes/No

Checklist for Upgrading

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

- If you intend to upgrade the Solaris software on a system through a `tip(1)` line, ensure that your window display is at least 80 columns wide and 24 rows long.
To determine the current dimensions of your `tip` window, use the `stty` command. For more information, see the man page, `stty(1)`.
- If the system is part of a network, verify that an Ethernet connector or similar network adapter is connected to your system.

- If you are using the Solaris Web Start program from the Solaris 9 Installation CD, verify that you have a 512-Mbyte slice on the disk.
- If you are using Solaris Live Upgrade, determine your resource requirements for creating a new boot environment and upgrading it. For detailed information, refer to Chapter 31.
- If you are using Solaris Live Upgrade with Solaris Volume Manager, turn root mirroring off. For detailed information, refer to “Guidelines for Selecting Slices for File Systems” on page 243.
- Verify that you have enough disk space. For more information, refer to Chapter 5.
- When you are using DVD media and are asked to boot from the ok prompt, always type the following command: **boot cdrom**.
- Review the *Solaris 9 Release Notes* and vendor release notes to ensure that the software you use is still supported in the new release.
- Review the *Solaris 9 Sun Hardware Platform Guide* to ensure that your hardware is supported.
- Review the documentation that came with your system to make sure your system and devices are supported by the Solaris release.
- Review vendor and third-party software documentation for additional upgrade instructions.
- Check for all of the available patches that you might need. The most recent patch list is provided at <http://sunsolve.sun.com>.
- Check the system for the existence of Prestoserve software. If you begin the upgrade process by shutting down the system with the `init 0` command, you might lose data. Refer to the Prestoserve documentation for shutdown instructions.

Worksheet for Upgrading

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

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

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

TABLE 6-2 Upgrade Worksheet

Info Needed to Install	Description/Example	Enter Your Answers Here:
Network	Is the system connected to a network?	Yes/No
DHCP	Can the system use Dynamic Host Configuration Protocol (DHCP) to configure its network interfaces?	Yes/No
Host Name	Host name that you choose for the system.	
IP Address	If you are not using DHCP, supply the IP address for the system. Example: 129.200.9.1	
Subnet	If you are not using DHCP, is the system part of a subnet? If yes, what is the netmask of the subnet? Example: 255.255.0.0	Yes/No
IPv6	Do you want to enable IPv6 on this machine?	Yes/No
Kerberos	Do you want to configure Kerberos security on this machine? If yes, gather this information: Default Realm: Administration Server: First KDC: (Optional) Additional KDCs:	Yes/No
Name Service	Which name service should this system use?	NIS+/NIS/DNS/LDAP/None
Domain Name	If the system uses a name service, supply the name of the domain in which the system resides.	

TABLE 6-2 Upgrade Worksheet (Continued)

Info Needed to Install	Description/Example	Enter Your Answers Here:
NIS+ and NIS	<p>Do you want to specify a name server or let the installation program find one?</p> <p>If you want to specify a name server, provide the following information.</p> <p style="text-align: right;">Server's host name:</p> <p style="text-align: right;">Server's IP Address:</p>	Specify One/Find One
DNS	<p>Provide IP addresses for the DNS server. You must enter at least one IP address, but you can enter up to three addresses.</p> <p style="text-align: right;">Server's IP Address(es):</p> <p>You can enter a list of domains to search when a DNS query is made.</p> <p style="text-align: right;">Search Domain:</p>	
LDAP	<p>Provide the following information about your LDAP profile.</p> <p style="text-align: right;">Profile Name:</p> <p style="text-align: right;">Profile Server:</p> <p style="text-align: right;">IP Address:</p>	
Default Router	<p>Do you want to specify a default IP router (gateway) or let the Solaris Web Start installation program find one?</p> <p>If you want to specify a default router, provide the following information.</p> <p style="text-align: right;">Router IP address:</p>	Specify One/Find One
Time Zone	How do you want to specify your default time zone?	<p>Geographic region</p> <p>Offset from GMT</p> <p>Time zone file</p>
Locales	For which geographic regions do you want to install support?	
Power Management	Do you want to use Power Management?	Yes/No

TABLE 6-2 Upgrade Worksheet (Continued)

Info Needed to Install	Description/Example	Enter Your Answers Here:
Proxy Server Configuration (only available in the Solaris Web Start program)	Do you have a direct connection to the Internet or do you need to use a proxy server to gain access to the Internet? If you use a proxy server, provide the following information. <p style="text-align: right;">Host: Port:</p>	Direct Connection/Proxy Server
Automatic reboot or CD/DVD ejection	Reboot automatically after software installation? Eject CD/DVD automatically after software installation?	Yes/No Yes/No
Disk Space Reallocation	Do you want the installation program to automatically re-lay out file systems on your disks? If yes, which file systems should be used for auto-layout? Example: /, /opt, /var If no, you must provide information for file system configuration.	Yes/No
64-bit	Do you want to install support for 64-bit applications?	Yes/No

Preconfiguring System Configuration Information (Tasks)

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

- “Advantages of Preconfiguring System Configuration Information” on page 47
- “Ways to Preconfigure System Configuration Information” on page 48
- “Preconfiguring With the `sysidcfg` File” on page 49
- “Preconfiguring With the Name Service” on page 54
- “Preconfiguring Power Management Information” on page 57

Advantages of Preconfiguring System Configuration Information

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

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

Ways to Preconfigure System Configuration Information

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

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

Use the following table to determine which method to use to preconfigure system configuration information.

TABLE 7-1 Methods to Preconfigure System Configuration Information

Preconfigurable System Information	Preconfigurable With the <code>sysidcfg</code> File?	Preconfigurable With the Name Service?
Name service	Yes	Yes
Domain name	Yes	No
Name server	Yes	No
Network interface	Yes	No
Host name	Yes ¹	Yes
Internet Protocol (IP) address	Yes ¹	Yes
Netmask	Yes	No
DHCP	Yes	No
IPv6	Yes	No
Default router	Yes	No
Root password	Yes	No
Security policy	Yes	No
Language (locale) in which to display the install program and desktop	Yes	Yes, if NIS or NIS + No, if DNS or LDAP
Terminal type	Yes	No
Time zone	Yes	Yes
Date and time	Yes	Yes

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

Preconfigurable System Information	Preconfigurable With the <code>sysidcfg</code> File?	Preconfigurable With the Name Service?
Power Management (autoshtutdown) ²	No	No

1. Because this information is system specific, edit the name service rather than create a different `sysidcfg` file for each system.
2. You cannot preconfigure this system configuration information through the `sysidcfg` file or the name service. “Preconfiguring Power Management Information” on page 57 contains details.

Preconfiguring With the `sysidcfg` File

You can specify a set of keywords in the `sysidcfg` file to preconfigure a system. The keywords are described in Table 7-2.

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

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

- NFS file system – If you put the `sysidcfg` file in a shared NFS file system, you must use the `-p` option of the `add_install_client(1M)` command when you set up the system to install from the network. The `-p` option specifies where the system can find the `sysidcfg` file when you install the Solaris 9 software.
- UFS diskette – Place the `sysidcfg` file in the root (`/`) directory on the diskette.

Note – If you are performing a custom JumpStart installation and you want to use a `sysidcfg` file on a diskette, you must place the `sysidcfg` file on the profile diskette. To create a profile diskette, see “Creating a Profile Diskette for Standalone Systems” on page 144.

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

Syntax Rules for the `sysidcfg` File

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

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

```
name_service=NIS {domain_name=marquee.central.sun.com
name_server=connor(129.152.112.3)}
```

Syntax Rule	Example
Independent keywords can be listed in any order.	<pre>pointer=MS-S display=ati {size=15-inch}</pre>
Keywords are not case sensitive.	<pre>TIMEZONE=US/Central</pre>
Enclose all dependent keywords in curly braces ({}) to tie them to their associated independent keyword.	<pre>name_service=NIS {domain_name=marquee.central.sun.com name_server=connor(129.152.112.3)}</pre>
You can optionally enclosed values in single (') or double quotes (").	<pre>network_interface='none'</pre>
Only one instance of a keyword is valid. However, if you specify the keyword more than once, only the first instance of the keyword is used.	<pre>network_interface=none network_interface=le0</pre>

`sysidcfg` File Keywords

The following table describes the keywords you can use in the `sysidcfg` file.

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

Configuration Information	Keywords	Values or Examples
Name service, domain name, name server	<code>name_service=NIS</code> , <code>NIS+</code> , <code>DNS</code> , <code>LDAP</code> , <code>NONE</code>	

TABLE 7-2 Keywords You Can Use in `sysidcfg` (Continued)

Configuration Information	Keywords	Values or Examples
	Options for NIS and NIS+: {domain_name=domain_name name_server=hostname(ip_address)}	name_service=NIS {domain_name=west.arp.com name_server=timber(129.221.2.1)} name_service=NIS+ {domain_name=west.arp.com. name_server=timber(129.221.2.1)}
	Options for DNS: {domain_name=domain_name name_server=ip_address,ip_address, ip_address (three maximum) search=domain_name, domain_name,domain_name, domain_name,domain_name, domain_name (six maximum, total length less than or equal to 250 characters)}	name_service=DNS {domain_name=west.arp.com name_server=10.0.1.10,10.0.1.20 search=arp.com,east.arp.com}
	Options for LDAP: {domain_name=domain_name profile=profile_name profile_server=ip_address}	name_service=LDAP {domain_name=west.arp.com profile=default profile_server=129.221.2.1}
Network interface, host name, Internet Protocol (IP) address, netmask, DHCP, IPv6	network_interface=NONE, PRIMARY, or value where: <ul style="list-style-type: none"> ■ NONE turns off networking ■ PRIMARY is the first up, non-loopback interface that is found on the system. The order is the same as with "ifconfig." If no interfaces are up, then the first non-loopback interface is used. If no non-loopback interfaces are found, then the system is set to NON-NETWORKED. ■ value specifies an interface such as le0 or hme0. 	
	If DHCP is to be used, specify: {dhcp protocol_ipv6=yes_or_no}	network_interface=primary {dhcp protocol_ipv6=yes}

Note – Choose only one value for name_service. Include either, both, or neither the domain_name and name_server keywords, as needed. If neither keyword is used, omit the curly braces {}.

TABLE 7-2 Keywords You Can Use in `sysidcfg` (Continued)

Configuration Information	Keywords	Values or Examples
	<p>If DHCP is <i>not</i> to be used, specify: <code>{hostname=host_name</code> <code>default_route=ip_address</code> <code>ip_address=ip_address</code> <code>netmask=netmask</code> <code>protocol_ipv6=yes_or_no}</code></p>	<pre>network_interface=le0 {hostname=feron default_route=129.146.88.1 ip_address=129.146.88.210 netmask=255.255.0.0 protocol_ipv6=no}</pre> <p>Note – Choose only one value for <code>network_interface</code>. Include any combination or none of the <code>hostname</code>, <code>ip_address</code>, and <code>netmask</code> keywords, as needed. If you do not use any of these keywords, omit the curly braces (<code>{}</code>).</p> <p>Note – If you do not use DHCP, you do not need to specify <code>protocol_ipv6</code> and <code>default_route</code>. But, a JumpStart installation requires <code>protocol_ipv6</code> to be specified now or you will be prompted interactively later.</p>
Root password	<code>root_password=root_password</code>	Encrypted from <code>/etc/shadow</code> .
Security policy	<p><code>security_policy=kerberos, NONE</code></p> <p>Options for Kerberos: <code>{default_realm=FQDN</code> <code>admin_server=FQDN kdc=FQDN1,</code> <code>FQDN2, FQDN3}</code> (FQDN is a fully qualified domain name)</p>	<pre>security_policy=kerberos {default_realm=Yoursite.COM admin_server=krbadmin.Yoursite.COM kdc=kdc1.Yoursite.COM, kdc2.Yoursite.COM}</pre> <p>Note – You can list a maximum of three key distribution centers (KDCs), but at least one is required.</p>
Language in which to display the install program and desktop	<code>system_locale=locale</code>	The <code>/usr/lib/locale</code> directory or Chapter 40 provides the valid locale values.
Terminal type	<code>terminal=terminal_type</code>	The subdirectories in the <code>/usr/share/lib/terminfo</code> directory provide the valid terminal values.
Time zone	<code>timezone=timezone</code>	The directories and files in the <code>/usr/share/lib/zoneinfo</code> directory provide the valid time zone values. The time zone value is the name of the path relative to the <code>/usr/share/lib/zoneinfo</code> directory. For example, the time zone value for mountain standard time in the United States is <code>US/Mountain</code> . The time zone value for Japan is <code>Japan</code> . You can also specify any valid Olson time zone.

TABLE 7-2 Keywords You Can Use in `sysidcfg` (Continued)

Configuration Information	Keywords	Values or Examples
Date and time	<code>timeserver=localhost, hostname, ip_addr</code>	If you specify <code>localhost</code> as the time server, the system's time is assumed to be correct. If you are not running a name service and you specify the <code>hostname</code> or <code>ip_addr</code> of a system, that system's time is used to set the time.

▼ To Create a `sysidcfg` Configuration File

1. Using a text editor, create a file called `sysidcfg`.
2. Type the `sysidcfg` keywords you want.
3. Save the `sysidcfg` file.

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

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

Example `sysidcfg` File

The following is an example of a `sysidcfg` file. The host names, IP addresses, and netmask of these systems have been preconfigured by editing the name service. Because all of the system configuration information is preconfigured in this file, you can use a custom JumpStart profile to perform a custom JumpStart installation.

```
system_locale=en_US
timezone=US/Central
terminal=sun-cmd
timeserver=localhost
name_service=NIS {domain_name=marquee.central.sun.com
                  name_server=connor(129.152.112.3)}
root_password=m4QPOWNY
```

Preconfiguring With the Name Service

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

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

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

- “To Preconfigure the Locale Using NIS” on page 54
- “To Preconfigure the Locale Using NIS+” on page 56

▼ To Preconfigure the Locale Using NIS

1. **Become superuser on the name server.**
2. **Change `/var/yp/Makefile` to add the local map.**
 - a. **Insert this shell procedure after the last `variable.time` shell procedure.**

```
locale.time: $(DIR)/locale
    -@if [ -f $(DIR)/locale ]; then \
        sed -e "/^#/d" -e s/#.*$$// $(DIR)/locale \
        | awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \
        | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/locale.byname; \
        touch locale.time; \
        echo "updated locale"; \
        if [ ! $(NOPUSH) ]; then \
            $(YPPUSH) locale.byname; \
            echo "pushed locale"; \
        else \
            : ; \
        fi \
    else \
```

```
        echo "couldn't find $(DIR)/locale"; \  
    fi
```

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

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

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

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

- d. Save the file.

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

```
locale domain_name
```

Or

```
locale system_name
```

Note – Chapter 40 contains a list of valid locales.

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

```
fr worknet.com
```

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

```
fr_BE charlie
```

Note – Locales are available on the Solaris 9 DVD or Solaris 9 Software 1 of 2 CD.

4. Make the maps:

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

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

▼ To Preconfigure the Locale Using NIS+

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

1. Log in to a name server as superuser or as a user in the NIS+ administration group.
2. Create the `locale` table:

```
# nistbladm -D access=og=rmcd,nw=r -c locale_tbl name=SI,nogw=
locale=,nogw= comment=,nogw= locale.org_dir.`nisdefaults -d`
```

3. Add needed entries to the `locale`.

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

<i>name</i>	Either the domain name or a specific system name for which you want to preconfigure a default locale.
<i>locale</i>	The locale you want to install on the system and use on the desktop after the system is rebooted. Chapter 40 contains a list of valid locales.
<i>comment</i>	The comment field. Use double quotation marks to begin and end comments that are longer than one word.

Note – Locales are available on the Solaris 9 DVD or Solaris 9 Software 1 of 2 CD.

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

Preconfiguring Power Management Information

You can use the *Power Management* software that is provided in the Solaris environment to automatically save the state of a system and turn it off after it is idle for 30 minutes. When you install the Solaris 9 software on a system that complies with Version 2 of the EPA's Energy Star guidelines, for example a sun4u system, the Power Management software is installed by default. You are then prompted after rebooting to enable or disable the Power Management software.

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

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

```
touch /a/autoshtutdown
```

Finish scripts are described in "Creating Finish Scripts" on page 161.

Upgrading the Solaris Operating Environment (Planning)

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

- “Upgrading” on page 59
- “Using Solaris Live Upgrade” on page 60
- “Using Custom JumpStart to Upgrade” on page 60
- “Upgrading With Disk Space Reallocation” on page 61
- “Backing Up Systems Before Upgrading” on page 62

Upgrading

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

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

```
$ uname -a
```

You can use any of the installation methods to upgrade the Solaris operating environment. For limitations on upgrading using Solaris Live Upgrade, see “Solaris Live Upgrade System Requirements” on page 239.

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

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

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

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

You can use the Patch Analyzer to determine which patches, if any, will be removed by upgrading to the Solaris 9 Update release. For detailed instructions about using the Patch Analyzer, refer to “Upgrading to a Solaris Update Release” on page 359.

Using Solaris Live Upgrade

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

For instructions on how to plan for and use Solaris Live Upgrade, see Chapter 29.

Using Custom JumpStart to Upgrade

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

You must test the custom JumpStart profile against the system’s disk configuration and currently installed software before you upgrade. Use the `pfinstall -D` command on the system that you are upgrading to test the profile. You cannot test an upgrade profile by using a disk configuration file. For more information about testing the upgrade option, refer to “Testing a Profile” on page 153.

Upgrading With Disk Space Reallocation

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

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

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

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

- If you are using the Solaris Web Start program, and auto-layout cannot determine how to reallocate the disk space, you must use the Solaris `suninstall` program or the custom JumpStart program to upgrade.
- If you are using the Solaris `suninstall` program, and auto-layout cannot determine how to reallocate disk space, you must specify the file systems that can be moved or changed and run auto-layout again.
- If you are using the custom JumpStart method to upgrade and you create an upgrade profile, disk space might be a concern. If the current file systems do not contain enough disk space for the upgrade, you can use the `backup_media` and `layout_constraint` keywords to reallocate disk space. For an example of how to use the `backup_media` and `layout_constraint` keywords in a profile, refer to Example 23-4.

Backing Up Systems Before Upgrading

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

Preparing to Install From the Network (Topics)

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

Chapter 10	Provides overview and planning information for installing Solaris software from an install server.
Chapter 11	Provides step-by-step instructions for copying Solaris software from DVD media to an install server.
Chapter 12	Provides step-by-step instructions for copying Solaris software from CD media to an install server.
Chapter 13	Describes commands to set up network installations.

Preparing to Install From the Network (Overview)

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

Planning for a Network Installation Introduction

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

Required Servers for Network Installation

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

- **Install server** – A networked system that contains the Solaris 9 disc images from which you can install Solaris 9 software on other systems on the network. You create an install server by copying the images from the following media:
 - Solaris 9 DVD
 - Solaris 9 Software 1 of 2 CD and Solaris 9 Software 2 of 2 CDAfter you copy the image from the Solaris 9 Software 1 of 2 and Solaris 9 Software 2 of 2 CDs, you can also copy the images from the Solaris 9 Installation CD and the Solaris 9 Languages CD as necessary for your

installation requirements.

You can enable a single install server to provide disc images for different Solaris releases by copying the images on to the install server's hard disk.

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

- "To Create an Install Server With DVD Media" on page 70
- "To Create an Install Server With CD Media" on page 81
- **Boot server** – A server system that provides client systems on the same network subnet with the information that they need to boot in order to install the operating environment. A boot server and install server are typically the same system. However, if the system on which the Solaris 9 software is to be installed is located in a different subnet than the install server and you are not using DHCP, a boot server is required on that subnet.

Note – When using DHCP, you do not need to create a separate boot server. For more information, see "Using DHCP to Provide Network Installation Parameters" on page 67.

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

- "Creating a Boot Server on a Subnet With a DVD Image" on page 73
- "Creating a Boot Server on a Subnet With a CD Image" on page 85
- **(Optional) Name server** – A system that manages a distributed network database, such as DNS, NIS, NIS+, or LDAP, that contains information about systems on the network.

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

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

Figure 10–1 illustrates the servers that are typically used for network installation.

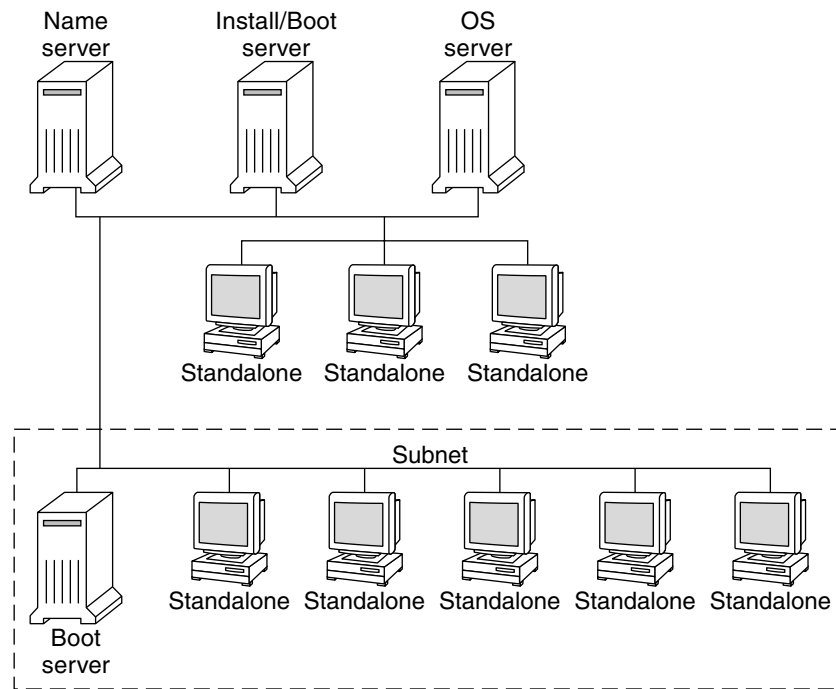


FIGURE 10-1 Network Installation Servers

Using DHCP to Provide Network Installation Parameters

When using Dynamic Host Configuration Protocol (DHCP), you do not need to create a separate boot server. After you have created the install server, you add clients to the network with the `add_install_client` command and the `-d` option. The `-d` option enables you to set up client systems for Solaris installation from the network by using DHCP. DHCP provides the network parameters that are necessary for installation. For information on DHCP options for installation parameters, see “Supporting Solaris Network Installation with the DHCP Service (Task Map)” in *System Administration Guide: IP Services*.

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

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

- "Task Map: Preparing to Install From the Network With DVD Media" on page 69
- "Creating an Install Server With DVD Media" on page 70
- "Creating a Boot Server on a Subnet With a DVD Image" on page 73
- "Adding Systems to Be Installed From the Network" on page 75

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

TABLE 11-1 Task Map: Setting Up an Install Server With DVD Media

Task	Description	For Instructions
Create an install server.	Use the <code>setup_install_server(1M)</code> command to copy the Solaris 9 DVD to the install server's hard disk.	"Creating an Install Server With DVD Media" on page 70

TABLE 11-1 Task Map: Setting Up an Install Server With DVD Media (Continued)

Task	Description	For Instructions
(Optional) Create boot servers.	If you want to install systems from the network that are not on the same subnet as the install server, you must create a boot server on the subnet to boot the systems. Use the <code>setup_install_server</code> command with the <code>-b</code> option to set up a boot server. If you are using Dynamic Host Configuration Protocol (DHCP), a boot server is not necessary.	"Creating a Boot Server on a Subnet With a DVD Image" on page 73
Add systems to be installed from the network.	Use the <code>add_install_client</code> command to set up each system that you want to install from the network. Each system that you want to install needs to find the install server, the boot server if required, and configuration information on the network.	"Adding Systems to Be Installed From the Network" on page 75

Creating an Install Server With DVD Media

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

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

▼ To Create an Install Server With DVD Media

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

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

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

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

2. Insert the Solaris 9 DVD in the system's drive.

3. Create a directory to contain the DVD image.

```
# mkdir -p install_dir_path
```

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

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

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

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

```
# ./setup_install_server install_dir_path
```

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

Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris 9 Software disc images. To determine available disk space, use the `df -k1` command.

6. Decide if you need to make the install server available for mounting.

- If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. Proceed to Step 7.
- If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must share the installation image with the boot server and start the NFS daemon. Continue.

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

```
# share | grep install_dir_path
```

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

- If the path to the install server's directory is displayed and anon=0 is displayed in the options, proceed to Step 7.
- If the path to the install server's directory is not displayed or you do not have anon=0 in the options, continue.

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

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

c. Make sure the install server's directory path is correctly shared.

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

- If nfsd daemon is not running, start it.

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

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

```
# shareall
```

7. Change directories to root (/).

```
# cd /
```

8. Eject the Solaris 9 DVD.

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

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



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

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

- If you are using DHCP or the install server is on the same subnet as the system to be installed, you do not need to create a boot server. Proceed to "Adding Systems to Be Installed From the Network" on page 75.

- If you are *not* using DHCP and the install server and the client is on a different subnet, you must create a boot server. Proceed to “Creating a Boot Server on a Subnet With a DVD Image” on page 73.

EXAMPLE 11-1 Creating an Install Server With DVD Media

The following example illustrates how to create an install server by copying the Solaris 9 DVD to the install server’s /export/home/s9dvdsparc directory:

```
# mkdir -p /export/home/s9dvdsparc
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
# ./setup_install_server /export/home/s9dvdsparc
```

If you need a separate boot server, add these steps:

```
Add the following path to the /etc/dfs/dfstab file and start NFS:
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/s9dvdsparc
# ps -ef | grep nfsd
If nfsd daemon is not running, start it.
# /etc/init.d/nfs.server start
If nfsd daemon is running, share the install server
# shareall
# cd /
```

In this example, each DVD is inserted and automatically mounted before each of the commands. After each command, the DVD is removed.

Creating a Boot Server on a Subnet With a DVD Image

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

- If you are using DHCP to set installation parameters or your install server or client are on the same subnet as the install server, you do not need a boot server. Proceed to “Adding Systems to Be Installed From the Network” on page 75.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.

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

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

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

2. **Mount the Solaris 9 DVD from the install server.**

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

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

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

```
# mkdir -p boot_dir_path
```

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

4. **Change to the `TOOLS` directory on the Solaris 9 DVD image.**

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

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

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

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

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

Note – The `setup_install_server` command indicates whether you have enough disk space available for the images. To determine available disk space, use the `df -k1` command.

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

```
# cd /
```

7. **Unmount the installation image.**

```
# umount /mnt
```

You are now ready to set up systems to be installed from the network. See “Adding Systems to Be Installed From the Network” on page 75.

EXAMPLE 11-2 Creating a Boot Server on a Subnet (DVDs)

The following example illustrates how to create a boot server on a subnet. These commands copy the boot software from the Solaris 9 DVD image to `/export/home/s9dvdsparc` on the local disk of a boot server named `crystal`.

```
# mount -F nfs -o ro crystal:/export/home/s9dvdsparc /mnt
# mkdir -p /export/home/s9dvdsparc
# cd /mnt/Solaris_9/Tools
# ./setup_install_server -b /export/home/s9dvdsparc
# cd /
# umount /mnt
```

In this example, each DVD is inserted and automatically mounted before each of the commands. After each command, the DVD is removed.

Adding Systems to Be Installed From the Network

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

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

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

- If you are using DHCP to set installation parameters, see Example 11-3.
- If your install server and client are on the same subnet, see Example 11-4.
- If your install server and your client are not on the same subnet and you are not using DHCP, see Example 11-5.

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

▼ To Add Systems to Be Installed From the Network With `add_install_client` From a Server

If you have a boot server, make sure you have shared the install server installation image and started the appropriate daemons. See Step 6.

1. On the install server or boot server, become superuser.
2. If you use the NIS, NIS+, DNS, or LDAP name service, verify that the following information about the system to be installed has been added to the name service.
 - Host name
 - IP address
 - Ethernet address

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

3. Change to the `Tools` directory on the Solaris 9 DVD image:

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

install_dir_path Specifies the path to the `Tools` directory

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

```
# ./add_install_client [-d] [-s install_server:install_dir_path] \  
[-c jumpstart_server:jumpstart_dir_path] [-p sysid_server:path] \  
[-t boot_image_path] client_name platform_group
```

<code>-d</code>	Specifies that the client is to use DHCP to obtain the network install parameters.
<code>-s install_server:install_dir_path</code>	Specifies the name and path to the install server. <ul style="list-style-type: none">■ <i>install_server</i> is the host name of the install server■ <i>install_dir_path</i> is the absolute path to the Solaris 9 DVD image
<code>-c jumpstart_server:jumpstart_dir_path</code>	Specifies a JumpStart directory for custom JumpStart installations. <i>jumpstart_server</i> is the host name of the server on which the JumpStart directory is located. <i>jumpstart_dir_path</i> is the absolute path to the JumpStart directory.

<code>-p sysid_server:path</code>	Specifies the path to the <code>sysidcfg</code> file for preconfiguring system information. <code>sysid_server</code> is either a valid host name or an IP address for the server that contains the file. <code>path</code> is the absolute path to the directory containing the <code>sysidcfg</code> file.
<code>-t boot_image_path</code>	Specifies the path to an alternate boot image if you want to use a boot image other than the one in the Tools directory on the Solaris 9 net installation image, CD, or DVD.
<code>client_name</code>	Is the name of the system to be installed from the network. This name is <i>not</i> the host name of the install server.
<code>platform_group</code>	Is the platform group of the system to be installed. For more information, see Chapter 39.

EXAMPLE 11-3 Adding an Install Client on an Install Server When Using DHCP (DVDs)

The following example illustrates how to add an install client when you are using DHCP to set installation parameters on the network. The install client is named `basil`, which is an Ultra™ 5 system. The file system `/export/home/s9dvdsparc/Solaris_9/Tools` contains the `add_install_client` command.

```
install_server# cd /export/home/s9dvdsparc/Solaris_9/Tools
install_server# ./add_install_client -d basil sun4u
```

EXAMPLE 11-4 Adding an Install Client That Is On the Same Subnet As Its Server (DVDs)

The following example illustrates how to add an install client that is on the same subnet as the install server. The install client is named `basil`, which is an Ultra™ 5 system. The file system `/export/home/s9dvdsparc/` contains the `add_install_client` command.

```
install_server# cd /export/home/s9dvdsparc/Solaris_9/Tools
install_server# ./add_install_client basil sun4u
```

EXAMPLE 11-5 Adding an Install Client to a Boot Server (DVDs)

The following example illustrates how to add an install client to a boot server. The install client is named `rose`, which is an Ultra™ 5 system. Run the command on the boot server. The `-s` option is used to specify an install server that is named `rosemary`, which contains a Solaris 9 DVD image in `/export/home/s9dvdsparc`.

```
boot_server# cd /export/home/s9dvdsparc/Solaris_9/Tools
boot_server# ./add_install_client -s rosemary:/export/home/s9dvdsparc rose sun4u
```


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

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

- "Task Map: Preparing to Install From the Network With CD Media" on page 80
- "Creating an Install Server With CD Media" on page 80
- "Creating a Boot Server on a Subnet With a CD Image" on page 85
- "Adding Systems to Be Installed From the Network" on page 87

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

TABLE 12-1 Task Map: Setting Up an Install Server With CD Media

Task	Description	For Instructions
Create an install server.	Use the <code>setup_install_server(1M)</code> command to copy the Solaris 9 Software 1 of 2 CD to the install server's hard disk. Use the <code>add_to_install_server(1M)</code> command to copy the Solaris 9 Software 2 of 2 CD and the Solaris 9 Languages CD to the install server's hard disk. Use the <code>modify_install_server(1M)</code> command to add the Solaris Web Start user interface software to the net installation image.	"Creating an Install Server With CD Media" on page 80
(Optional) Create boot servers.	If you want to install systems from the network that are not on the same subnet as the install server, you must create a boot server on the subnet to boot the systems. If you are using Dynamic Host Configuration Protocol (DHCP), a boot server is not necessary.	"Creating a Boot Server on a Subnet With a CD Image" on page 85
Add systems to be installed from the network.	Use the <code>add_install_client</code> command to set up each system that you want to install from the network. Each system that you want to install needs to find the install server, the boot server if required, and configuration information on the network.	"Adding Systems to Be Installed From the Network" on page 87

Creating an Install Server With CD Media

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

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

▼ To Create an Install Server With CD Media

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

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

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

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

2. Insert the Solaris 9 Software 1 of 2 CD in the system's drive.

3. Create a directory for the CD image.

```
# mkdir -p install_dir_path
```

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

4. Change to the `TOOLS` directory on the mounted disc.

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

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

```
# ./setup_install_server install_dir_path
```

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

Note – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris 9 Software disc images. To determine available disk space, use the `df -k1` command.

6. Decide if you need make the install server available for mounting.

- If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. Proceed to Step 7.
- If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must share the installation image with the boot server and start the NFS daemon. Continue.

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

```
# share | grep install_dir_path
```

install_dir_path Specifies the path to boot server installation image

- If the path to the install server's directory is displayed and `anon=0` is displayed in the options, proceed to Step 7.
- If the path to the install server's directory is not displayed or you do not have `anon=0` in the options, continue.

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

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

c. Make sure the install server's directory path is correctly shared.

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

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

```
# /etc/init.d/nfs.server start
```
- If `nfsd` daemon is running, share the install server.

```
# shareall
```

7. Change directories to root (/).

```
# cd /
```

8. Eject the Solaris 9 Software 1 of 2 CD.

9. Insert the Solaris 9 Software 2 of 2 CD in the system's CD-ROM drive.

10. Change to the `Tools` directory on the mounted CD:

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

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

```
# ./add_to_install_server install_dir_path
```

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

12. Change directories to root (/).

```
# cd /
```

13. Eject the Solaris 9 Software 2 of 2 CD.

14. Insert the Solaris 9 Languages CD in the system's CD-ROM drive.

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

```
# cd /cdrom/cdrom0/sol_9_lang_sparc/Tools
```

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

```
# ./add_to_install_server install_dir_path
```

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

17. Change directories to root (/).

```
# cd /
```

18. Decide if you want to enable users to use the Solaris Web Start installation method to boot a system and install the Solaris 9 software from a network.

- If no, eject the Solaris 9 Languages CD and proceed to Step 22.
- If yes, eject the Solaris 9 Languages CD and continue.

19. Insert the Solaris 9 Installation CD into the system's CD-ROM drive.

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

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

21. Copy the Solaris Web Start interface software to the install server.

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

<code>-p</code>	Preservers the existing image's miniroot in <code>install_dir_path/Solaris_9/Tools/Boot.orig</code>
<code>install_dir_path</code>	Specifies the directory where the Solaris Web Start interface is to be copied
<code>installer_miniroot_path</code>	Specifies the directory on the CD from which the Solaris Web Start interface is to be copied

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

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



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

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

- If you are using DHCP or the install server is on the same subnet as the system to be installed, you do not need to create a boot server. Proceed to "Adding Systems to Be Installed From the Network" on page 75.
- If you are *not* using DHCP and the install server and the client is on a different subnet, you must create a boot server. Proceed to "Creating a Boot Server on a Subnet With a DVD Image" on page 73.

EXAMPLE 12-1 Creating an Install Server With CD Media

The following example illustrates how to create an install server by copying the following CDs to the install server's `/export/home/s9cdsparc` directory:

- Solaris 9 Software 1 of 2 CD
- Solaris 9 Software 2 of 2 CD
- Solaris 9 Languages CD

```
# mkdir -p /export/home/s9cdsparc
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
# ./setup_install_server /export/home/s9cdsparc
```

If you have a separate boot server, add these steps:

```
Add the following path to the /etc/dfs/dfstab file:
share -F nfs -o ro,anon=0 -d "install server directory" \
/export/home/s9cdsparc
# ps -ef | grep nfsd
```

EXAMPLE 12-1 Creating an Install Server With CD Media (Continued)

```
If nfsd daemon is not running, start it.  
# /etc/init.d/nfs.server start  
If nfsd daemon is running, share it.  
# shareall  
Continue with the following steps.
```

If you do not need a boot server or have added the steps for a separate boot server, continue:

```
# cd /  
# cd /cdrom/cdrom0/Solaris_9/Tools  
# ./add_to_install_server /export/home/s9cdsparc  
# cd /  
# cd /cdrom/cdrom0/sol_9_lang_sparc/Tools  
# ./add_to_install_server /export/home/s9cdsparc  
# cd /  
# cd /cdrom/cdrom0/s0  
# ./modify_install_server -p /export/home/s9cdsparc /cdrom/cdrom0/s0
```

In this example, each CD is inserted and automatically mounted before each of the commands. After each command, the CD is removed.

Creating a Boot Server on a Subnet With a CD Image

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

- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a boot server. Proceed to “Adding Systems to Be Installed From the Network” on page 87.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.

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

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

The system must include a local CD-ROM drive or have access to the remote Solaris 9 disc images, which are normally on the install server. If you use a name service, the system should be in the name service. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. Mount the Solaris 9 Software 1 of 2 CD image from the install server.

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

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

3. Create a directory for the boot image.

```
# mkdir -p boot_dir_path
```

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

4. Change to the `TOOLS` directory on the Solaris 9 Software 1 of 2 CD image.

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

5. Copy the boot software to the boot server.

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

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

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

Note – The `setup_install_server` command indicates whether you have enough disk space available for the images. To determine available disk space, use the `df -k1` command.

6. Change directories to root (/).

```
# cd /
```

7. Unmount the installation image.

```
# umount /mnt
```

You are now ready to set up systems to be installed from the network. See “Adding Systems to Be Installed From the Network” on page 87.

EXAMPLE 12-2 Creating a Boot Server on a Subnet With CD Media

The following example illustrates how to create a boot server on a subnet. These commands copy the boot software from the Solaris 9 Software 1 of 2 CD image to `/export/install/boot` on the system's local disk.

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

In this example, the disc is inserted and automatically mounted before the command. After the command, the disc is removed.

Adding Systems to Be Installed From the Network

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

- An install server
- A boot server if it is required
- The `sysidcfg` file if you use a `sysidcfg` file to preconfigure system information
- A name server if you use a name service to preconfigure system information
- The profile in the JumpStart directory on the profile server if you are using the custom JumpStart installation method

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

- If you are using DHCP to set installation parameters, see Example 12-3.
- If your install server and client are on the same subnet, see Example 12-4.
- If your install server and your client are not on the same subnet and you are not using DHCP, see Example 12-5.

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

▼ To Add Systems to Be Installed From the Network With `add_install_client` From a Server

If you have a boot server, make sure you have shared the install server installation image. See the procedure “To Create an Install Server,” Step 6.

1. On the install server or boot server, become superuser.
2. If you use the NIS, NIS+, DNS, or LDAP name service, verify that the following information about the system to be installed has been added to the name service:

- Host name
- IP address
- Ethernet address

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

3. Change to the `TOOLS` directory on the Solaris 9 CD image on the install server:

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

install_dir_path Specifies the path to the `TOOLS` directory

4. Set up the client system to be installed from the network.

```
# ./add_install_client [-d] [-s install_server:install_dir_path] \  
[-c jumpstart_server:jumpstart_dir_path] [-p server:path] \  
[-t boot_image_path] client_name platform_group
```

<code>-d</code>	Specifies that the client is to use DHCP to obtain the network install parameters.
<code>-s install_server:install_dir_path</code>	Specifies the name and path to the install server. <ul style="list-style-type: none">■ <i>install_server</i> is the host name of the install server■ <i>install_dir_path</i> is the absolute path to the Solaris 9 CD image
<code>-c jumpstart_server:jumpstart_dir_path</code>	Specifies a JumpStart directory for custom JumpStart installations. <i>jumpstart_server</i> is the host name of the server on which the JumpStart directory is located. <i>jumpstart_dir_path</i> is the absolute path to the JumpStart directory.

<code>-p <i>sysid_server</i> : <i>path</i></code>	Specifies the path to the <code>sysidcfg</code> file for preconfiguring system information. <i>sysid_server</i> is either a valid host name or an IP address for the server that contains the file. <i>path</i> is the absolute path to the directory containing the <code>sysidcfg</code> file.
<code>-t <i>boot_image_path</i></code>	Specifies the path to an alternate boot image if you want to use a boot image other than the one in the Tools directory on the Solaris 9 net installation image, CD, or DVD.
<code><i>client_name</i></code>	Is the name of the system to be installed from the network. This name is <i>not</i> the host name of the install server.
<code><i>platform_group</i></code>	Is the platform group of the system to be installed. A detailed list of platform groups appears in Chapter 39.

EXAMPLE 12-3 Adding an Install Client on an Install Server When Using DHCP (CDs)

The following example illustrates how to add an install client when you are using DHCP to set installation parameters on the network. The install client is named `basil`, which is an Ultra™ 5 system. The file system `/export/home/s9cdsparc/Solaris_9/Tools` contains the `add_install_client` command.

```
install_server# cd /export/home/s9cdsparc/Solaris_9/Tools
install_server# ./add_install_client -d basil sun4u
```

EXAMPLE 12-4 Adding an Install Client That Is on the Same Subnet as Its Server (CDs)

The following example illustrates how to add an install client that is on the same subnet as the install server. The install client is named `basil`, which is an Ultra™ 5 system. The file system `/export/home/s9cdsparc/Solaris_9/Tools` contains the `add_install_client` command.

```
install_server# cd /export/home/s9cdsparc/Solaris_9/Tools
install_server# ./add_install_client basil sun4u
```

EXAMPLE 12-5 Adding an Install Client to a Boot Server (CDs)

The following example illustrates how to add an install client to a boot server. The install client is named `rose`, which is an Ultra™ 5 system. Run the command on the boot server. The `-s` option is used to specify an install server that is named `rosemary`, which contains a Solaris 9 CD image in `/export/home/s9cdsparc`.

```
boot_server# cd /export/home/s9cdsparc/Solaris_9/Tools
boot_server# ./add_install_client -s rosemary:/export/home/s9cdsparc rose sun4u
```


Preparing to Install From the Network (Command Reference)

This chapter lists the commands used to set up network installations.

TABLE 13-1 Network Installation Commands

Command	Description
<code>add_install_client</code>	A command that adds network installation information about a system to an install server or boot server in order to install from the network. The <code>add_install_client(1M)</code> man page contains more information.
<code>setup_install_server</code>	A script that copies the Solaris 9 DVDs or CDs to an install server's local disk or copies the boot software to a boot server. The <code>setup_install_server(1M)</code> man page contains more information.
<code>add_to_install_server</code> (CD media only)	A script that copies additional packages within a product tree on the CDs to the local disk on an existing install server. The <code>add_to_install_server(1M)</code> man page contains more information.
<code>modify_install_server</code> (CD media only)	A script that adds the Solaris Web Start user interface software to the Solaris 9 CD images on an existing boot server. This script enables you to use the Solaris Web Start program to boot a system and install the Solaris 9 software from a network. The <code>modify_install_server(1M)</code> man page contains more information. A system with 64 Mbytes or less RAM will not have enough memory to install a machine by using Solaris Web Start.
<code>mount</code>	A command that enables the mounting of file systems and shows mounted file systems, including the file system on the Solaris 9 DVD or Solaris 9 Software and Solaris 9 Languages CDs. The <code>mount(1M)</code> man page contains more information.

TABLE 13-1 Network Installation Commands (Continued)

Command	Description
<code>uname -i</code>	A command for determining a system's platform name, for example SUNW, SPARCstation-5. You might need the system's platform name when you install the Solaris software. The <code>uname(1)</code> man page contains more information.
<code>patchadd -C net_install_image</code>	<p>A command to add patches to the files that are located in the miniroot, <code>Solaris_9/Tools/Boot</code>, on a net installation image of a DVD or CD that is created by <code>setup_install_server</code>. This facility enables you to patch Solaris installation commands and other miniroot-specific commands. <code>net_install_image</code> is the absolute path name of the net installation image. The <code>patchadd(1M)</code> man page contains more information.</p> <p>Caution – Don't use the <code>patchadd -C</code> unless you have read the Patch Readme instructions or have contacted your local Sun support office.</p>
<code>reset</code>	An Open Boot prom command for resetting the system and rebooting the machine. Or, if you boot and see a series of error messages about I/O interrupts, press the Stop and A keys at the same time, and then type <code>reset</code> at the <code>ok</code> or <code>> PROM</code> prompt.
<code>banner</code>	An Open Boot prom command that displays system information, such as model name, Ethernet address, and memory installed. You can issue this command only at the <code>ok</code> or <code>> PROM</code> prompt.

Using the Solaris Web Start Program (Tasks)

This chapter explains how to use the Solaris Web Start program on the Solaris 9 DVD or Solaris 9 Installation CD to install or upgrade Solaris software.

Note – If you want to install the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive, you can use a DVD-ROM or CD-ROM drive that is attached to another machine. For detailed instructions, refer to Appendix B.

This chapter contains the following topics.

- “Solaris Web Start Program GUI or CLI” on page 93
- “Performing an Installation or Upgrade With the Solaris Web Start Program” on page 94
- “Solaris Web Start Post-Installation and Upgrade Tasks” on page 98

Solaris Web Start Program GUI or CLI

You can run the Solaris Web Start program with a GUI or with a CLI.

- GUI – Requires a local or remote DVD-ROM or CD-ROM drive or network connection, video adapter, keyboard, monitor, and enough memory.
- CLI – Requires a local or remote DVD-ROM or CD-ROM drive or network connection, keyboard, and monitor. You can run the Solaris Web Start installation CLI with the `tip` command. For more information, see the man page, `tip(1)`.

If the Solaris Web Start program detects a video adapter for the system, it automatically displays the GUI. If the Solaris Web Start program does not detect a video adapter, it automatically displays the CLI. The content and sequence of instructions in both the GUI and the CUI are generally the same.

Performing an Installation or Upgrade With the Solaris Web Start Program

You can either install or upgrade the Solaris operating environment by using the Solaris Web Start program, which is on the Solaris 9 DVD or the Solaris 9 Installation CD.

Note – If you are installing from the Solaris 9 Installation CD, refer to “Requirements When Using the Solaris 9 Installation CD” on page 24 for special requirements.

For the installation, ensure that you have the following:

- For a DVD installation, the Solaris 9 DVD
- For a CD installation:
 - Solaris 9 Installation CD.
 - Solaris 9 Software 1 of 2 CD.
 - Solaris 9 Software 2 of 2 CD – The installation program prompts you for this CD if necessary.
 - Solaris 9 Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

Task Map: Performing a Solaris Web Start Installation

TABLE 14-1 Performing a Solaris Web Start Installation Task Map

Task	Description	For Instructions
Verify system requirements.	Verify that your system meets the requirements to install or upgrade with the Solaris Web Start program.	“System Requirements” on page 24

TABLE 14-1 Performing a Solaris Web Start Installation Task Map (Continued)

Task	Description	For Instructions
Gather the necessary information.	Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.	Chapter 6
(Optional) Preconfigure system configuration information.	Use the <code>sysidcfg</code> file or the name service to preconfigure installation information (for example, <code>locale</code>) for a system. Preconfiguring system information prevents the installation program from prompting you during the installation.	Chapter 7
(Upgrade only) Prepare to upgrade the system.	Back up the system.	<i>System Administration Guide: Basic Administration</i>
(Optional) Set up the system to install from the network.	To install a system from a remote DVD or CD net installation image, you need to set up the system to boot and install from an install server or boot server.	Chapter 12
Install or upgrade.	Boot the system and follow the prompts to install or upgrade the Solaris software.	"To Perform an Installation or Upgrade With the Solaris Web Start Program" on page 95
(Upgrade only) Perform post-upgrade tasks.	Correct any local modifications that were lost during the upgrade.	"To Correct Local Modifications After Upgrading" on page 98

▼ To Perform an Installation or Upgrade With the Solaris Web Start Program

1. Decide if you want to install the software by using the DVD-ROM or CD-ROM drive or by using a net image.

- If you're using a DVD-ROM or CD-ROM drive, insert the Solaris 9 DVD or Solaris 9 Installation CD.
- If you're using a net installation image, change directories to where the installation media is located. You might need to contact your network administrator for the location. The following command is an example.

```
% cd /net/install-svr/export/s9/sparc
```

2. Boot the system.

- If the system is new, out-of-the-box, turn on the system.
- If you want to install or upgrade an existing system, shut down the system.
 - To boot from the local DVD or CD, type:


```
ok boot cdrom [- nowin]
```
 - To boot from an install server on a network, type:

```
ok boot net [- nowin]
```

nowin Specifies to run the Solaris Web Start program in CLI mode.

The Solaris Web Start installation begins.

3. If you are prompted, answer the system configuration questions.

- If you preconfigured all of the system configuration information, the Solaris Web Start program does not prompt you to enter any configuration information. Proceed to Step 7.
- If you did not preconfigure the system configuration information, use the “Worksheet for Installation” on page 38 or the “Worksheet for Upgrading” on page 42 to help you answer the system configuration questions.

If you are using the GUI, after you confirm the system configuration information, the Solaris Web Start Installation Kiosk and Welcome to Solaris dialog box appear. If your system has insufficient memory, the Kiosk does not display.



FIGURE 14-1 Solaris Web Start Kiosk Menu

You can click on any link in the Kiosk menu.

Note – In some situations, the Kiosk might cover a dialog box. To display a hidden dialog box, from the Kiosk menu, choose Send Kiosk to Background.

The Installer Questions screen appears.

4. **Decide if you want to reboot the system automatically and if you want to automatically eject the disc. Click NEXT.**
The Specify Media screen appears.
5. **Specify the media you are using to install: CD or DVD, Network, HTTP, FTP, or Local tape.**
6. **Decide if you want to perform an initial installation or an upgrade.**

The Solaris Web Start program determines if the system can be upgraded. You must have an existing Solaris root (/) file system. If you are upgrading by using the Solaris 9 Installation CD, you must have a 512-Mbyte slice. The Solaris Web Start program detects the necessary conditions and then upgrades the system.

7. Follow the instructions on the screen to install the Solaris software and any additional software on the system.

When the Solaris Web Start program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

- /var/sadm/system/logs
- /var/sadm/install/logs

8. If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to “To Correct Local Modifications After Upgrading” on page 98.

Solaris Web Start Post-Installation and Upgrade Tasks

After you install or upgrade the Solaris operating environment, you might need to perform these tasks.

▼ To Correct Local Modifications After Upgrading

When you upgrade, the Solaris Web Start program merges local software modifications of the existing system with the new Solaris software. However, in some situations, merging is not possible. After you finish upgrading a system, you might need to correct some local modifications that were not preserved.



Caution – Your system might not boot if you do not correct the local modifications that were not preserved.

1. Review the contents of the following file to determine whether you need to correct local modifications that the Solaris Web Start program could not preserve.

/a/var/sadm/system/data/upgrade_cleanup

2. Correct any local modifications that were not preserved.

3. Reboot the system.

```
# reboot
```


Using the Solaris `suninstall` Program (Tasks)

This chapter explains how to use the Solaris `suninstall` program on the Solaris 9 Software 1 of 2 CD to install or upgrade the Solaris software.

- “Solaris `suninstall` Program ” on page 101
- “Performing an Installation or Upgrade With the Solaris `suninstall` Program” on page 101
- “Solaris `suninstall` Program Post-Upgrade Task” on page 104

Solaris `suninstall` Program

You can run the Solaris `suninstall` program with a CLI. The Solaris `suninstall` program requires a local or remote CD-ROM drive or network connection, keyboard, and monitor. You can run the Solaris `suninstall` program with the `tip` command. For more information, see the man page, `tip(1)`.

Note – To navigate through the text, use Control-U to move up a page and Control-D to move down a page.

Performing an Installation or Upgrade With the Solaris `suninstall` Program

You can either install or upgrade the Solaris operating environment on your system by using the Solaris `suninstall` program on the Solaris 9 Software 1 of 2 CD. You cannot install any additional software with the Solaris `suninstall` program.

Ensure that you have the following CDs:

- Solaris 9 Software 1 of 2 CD.
- Solaris 9 Software 2 of 2 CD – The installation program prompts you for this CD if necessary.
- Solaris 9 Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

Task Map: Performing a Solaris `suninstall` Program Installation

TABLE 15-1 Task Map: Performing a Solaris `suninstall` Program Installation

Task	Description	For Instructions
Gather the necessary information.	Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.	Chapter 6
(Upgrade only) Prepare to upgrade the system.	Back up the system.	<i>System Administration Guide: Basic Administration</i>
(Optional) Preconfigure system configuration information.	Use the <code>sysidcfg</code> file or the name service to preconfigure installation information for a system, for example <code>locale</code> . Preconfiguring system information prevents the installation program from prompting you to supply the information during the installation.	Chapter 7
(Optional) Set up the system to install from the network.	To install a system from a remote Solaris 9 Software CD image, you need to set up the system to boot and install from an install server or boot server.	Chapter 12
Install or upgrade.	Boot the system and follow the prompts to install or upgrade the Solaris software.	“To Perform an Installation or Upgrade With the Solaris <code>suninstall</code> Program” on page 102
(Upgrade only) Perform post-upgrade task.	Correct any local modifications that were lost during the upgrade.	“Solaris <code>suninstall</code> Program Post-Upgrade Task” on page 104

▼ To Perform an Installation or Upgrade With the Solaris `suninstall` Program

1. **Decide if you want to install the software by using a CD-ROM drive or by using a net image.**
 - If you’re using a CD-ROM drive, insert the Solaris 9 Software 1 of 2 CD

- If you're using a net installation image, change directories to where the installation media is located, as in the following example. You might need to check with your network administrator for the location. The following command is an example.

```
% cd /net/install-svr/export/s9/sparc
```

2. Boot the system.

- If the system is new, out-of-the-box, turn on the system.
- If you want to install or upgrade an existing system, shut down the system.

- To boot from the local CD, type:

```
ok boot cdrom
```

- To boot from an install server on a network, type:

```
ok boot net
```

The Solaris `suninstall` program begins.

3. If you are prompted, answer the system configuration questions.

If you preconfigured all of the system configuration information, the Solaris `suninstall` program does not prompt you to enter any configuration information.

If you did not preconfigure the system configuration information, use the "Worksheet for Installation" on page 38 or the "Worksheet for Upgrading" on page 42 to help you answer the system configuration questions.

4. Follow the instructions on the screen to install the Solaris software on the system.

When the Solaris `suninstall` program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.

After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

- `/var/sadm/system/logs`
- `/var/sadm/install/logs`

5. If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to "Solaris `suninstall` Program Post-Upgrade Task" on page 104.

6. (Optional) To install additional software, refer to *System Administration Guide: Basic Administration*.

Solaris `suninstall` Program Post-Upgrade Task

When you upgrade, the Solaris `suninstall` program merges local software modifications of the existing system with the new Solaris software. However, in some situations, merging is not possible. After you finish upgrading a system, you might need to correct some local modifications that were not preserved.



Caution – Your system might not boot if you do not correct the local modifications that were not preserved.

▼ To Correct Local Modifications After Upgrading

1. **Review the contents of the following file to determine whether you need to correct local modifications that the Solaris `suninstall` program could not preserve.**

`/a/var/sadm/system/data/upgrade_cleanup`

2. **Correct any local modifications that were not preserved.**
3. **Reboot the system.**

```
# reboot
```


Web Start Flash Installation Feature (Topics)

This section provides instructions for creating Web Start Flash archives and using Web Start Flash archives to install the Solaris operating environment on multiple systems.

Chapter 17	Provides overview and planning information on creating and installing Web Start Flash archives.
Chapter 18	Provides step-by-step instructions for creating Web Start Flash archives.
Chapter 19	Provides step-by-step instructions for using Web Start Flash archives to install systems.
Chapter 20	Describes syntax and options for the <code>flarcreate</code> and <code>flar</code> commands. Describes keywords for Web Start Flash archive information.

Web Start Flash (Overview and Planning)

This chapter provides an introduction to the Web Start Flash installation feature. Also included in the chapter is information necessary for planning a Web Start Flash installation in your environment.

- “Web Start Flash Introduction” on page 107
- “Planning Your Web Start Flash Installation” on page 108

Web Start Flash Introduction

The Web Start Flash installation feature enables you to create a single reference installation of the Solaris operating environment on a system, which is called the master system. Then you can replicate that installation on a number of systems, which are called clone systems. Installing clone systems with the Web Start Flash installation method is a three-part process.

1. Install the master system. You select a system and use any of the Solaris installation methods to install the Solaris operating environment and any other software.
2. Create the Web Start Flash archive. The Web Start Flash archive contains a copy of all of the files on the master system.
3. Install the Web Start Flash archive on clone systems. When you install the Web Start Flash archive on a system, all of the files in the archive are copied to that system. The newly installed system now has the same installation configuration as the original master system, thus it is called a clone system.

Note – You cannot upgrade a system that is running the Solaris operating environment by using the Web Start Flash installation feature. You can only perform an initial installation.

Planning Your Web Start Flash Installation

Before you create and install a Web Start Flash archive, you must make some decisions about how you want to install the Solaris operating environment on your systems.

Designing the Installation of the Master System

The first task in the Web Start Flash installation process is to install a system, the master system, with the configuration that you want each of the clone systems to have. You can use any of the Solaris installation methods to install a subset or a complete installation of the Solaris operating environment on the master system. After you complete the installation, you can add or remove software or modify any configuration files.

The master system and the clone systems must have the same kernel architectures. For example, you can use a Web Start Flash archive that was created from a master system that has a sun4u architecture only to install other systems with a sun4u architecture.

You must install the master system with the exact configuration that you want on each of the systems that you are installing with the Web Start Flash archive that was created from this master system. The decisions you make when you design the installation of the master system depend on the following:

- The software you want to install on the clone systems
- Peripheral devices that are connected to the master system and the clone systems
- The architecture of the master system and the clone systems

Customizing the Solaris Installation on the Master System

After you install the Solaris operating environment on the master system by using any of the Solaris installation methods, you can add or delete software and modify system configuration information as necessary.

- Delete software. You can remove software that you determine is not necessary to install on the clone systems. To see a list of software that is installed on the master system, use the Product Registry. For detailed instructions, refer to *System Administration Guide: Basic Administration*.
- Add software. You can install software that is included in the Solaris release and software that is not delivered as part of the Solaris operating environment. All of the software that you install on the master system is included in the Web Start Flash archive and is installed on the clone systems.

- Modify configuration files. You can alter configuration files on the master system. For example, you can modify the `/etc/inet/inetd.conf` file to restrict the daemons that the system runs. All of the modifications that you make are saved as part of the Web Start Flash archive and are installed on the clone systems.

Note – After you install the Web Start Flash archive on a clone system, some host-specific files are deleted and re-created for the clone machine. The installation program uses the `sys-unconfig(1M)` command and the `sysidtool(1M)` programs to delete and re-create the host-specific network configuration files. The files that are re-created include such files as `/etc/hosts`, `/etc/defaultrouter`, and `/etc/defaultdomain`.

Supporting Peripheral Devices Not Found on the Master System

You might be installing a master system that has different peripheral devices than the clone systems. If you install the master system with the Core, End User, Developer, or Entire Software Group, the master system supports only the peripheral devices that are attached to the master system at the time of installation.

For example, if you install the Entire Software Group on a master system that has a cg6 frame buffer, the installation contains support for only the cg6 frame buffer. Consequently, you can only install the archive that you create from this system on clone systems that have either the cg6 frame buffer or no frame buffers. If you use the archive to install a clone system with an Elite 3D frame buffer, the Elite 3D will be unusable because the required drivers will not be installed.

You might have clone systems that have peripheral devices that the master system does not have. You can install support for these peripheral devices on the master system even though the master system does not have the devices. The Web Start Flash archive you create from this master system contains support for the peripheral devices on the clone systems.

If you plan to install clone systems that have different peripherals from the master system, you can install support for those peripherals on the master system in one of the following ways.

- Install the Entire Plus OEM Software Group – The Entire Plus OEM Software Group is the largest Software Group available and contains every package found in the Solaris operating environment. If you install the Entire Plus OEM Software group on the master system, the master system includes all of the drivers that are included with the Solaris release. A Web Start Flash archive that is created from a master system that you install with the Entire Plus OEM Software Group works on any clone system that has peripheral devices supported by the installed release of the Solaris operating environment.

Installing master systems with the Entire Plus OEM Software Group guarantees compatibility with other peripheral configurations. However, the Entire Plus OEM Software Group requires over 1 Gbyte of disk space. The clone systems might not have the space that is required to install the Entire Plus OEM Software Group.

- Install selected packages – When you install the master system, you can install only the packages that you need for the master system and the clone systems. By selecting specific packages, you can install only support for the peripherals that you know exist on the master system or clone systems.

Planning the Creation of a Web Start Flash Archive

After you install the master system, the next task in the Web Start Flash installation process is to create a Web Start Flash archive. All of the files on the master system are copied to a Web Start Flash archive along with various pieces of identification information. You can create a Web Start Flash archive while the master system is running in multiuser mode or single-user mode. You can also create a Web Start Flash archive after you boot from one of the following:

- Solaris 9 DVD
- Solaris 9 Software 1 of 2 CD
- An image of the Solaris 9 Software and the Solaris 9 Languages CDs

Create the archive when the system is in as static a state as possible.

Archive Identification Information

A Web Start Flash archive contains archive identification information in addition to the actual files from the master system that will be installed on the clone systems. You are required to specify a name for the Web Start Flash archive. Other information that you can specify about the archive includes:

- The author of the archive
- The date the archive was created
- The name of the master system that you used to create the archive

For a complete list of the archive identification information that you can specify, refer to “Identification Section Keywords” on page 124.

Use the `flar` command to retrieve information about the archive. For detailed instructions, refer to “`flar`” on page 128.

Where to Store the Web Start Flash Archive

After you create the Web Start Flash archive, you can save the archive on the hard disk of the master system or on a tape. After you save the archive, you can copy it to any file system or media that you choose.

- Network File System (NFS)
- HTTP server
- FTP server
- Tape
- CD, DVD
- Diskette
- Local drive of clone system that you want to install

Compressing the Archive

When you create the Web Start Flash archive, you can specify that the archive be saved as a compressed file by using the `compress (1)` utility. An archive that is compressed requires smaller disk storage space and creates less congestion when you install the archive over a network.

Planning the Installation of Web Start Flash Archives

The final task in the Web Start Flash installation process is to install Web Start Flash archives on clone systems.

Deciding How to Install Web Start Flash Archives

You can use any of the Solaris installation methods to install Web Start Flash archives on clone systems.

The Solaris Web Start program on the Solaris 9 DVD or Solaris 9 Installation CD enables you to install Web Start Flash archives that are stored on the following:

- Disc (DVD or CD)
- NFS server
- HTTP server
- FTP server
- Local tape

The Solaris `suninstall` program on the Solaris 9 Software 1 of 2 CD enables you to install Web Start Flash archives that are stored on the following:

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

The custom JumpStart installation program enables you to install Web Start Flash archives that are stored on the following:

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

Solaris Live Upgrade enables you to install Web Start Flash archives that are stored on the following:

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

Installing Layered Web Start Flash Archives

The Web Start Flash installation feature provides the ability to layer Web Start Flash archives. You can create partial Web Start Flash archives to install in a variety of ways.

For example, you can create one archive that contains the Solaris operating environment files, a second archive that contains the files necessary to run a Web server, and a third archive that contains the files for an NFS server. You can install the first and second archives to a system in order to create a Web server. You can install the first and third archives to another system in order to create an NFS server.

By using layered archives, you can increase the flexibility of the Web Start Flash installation while you reduce the disk space that is required to store Web Start Flash archives. When you install layered archives to a clone system, one of the archives must contain the Solaris operating environment.

Note – If you use layered Web Start Flash archives to install additional software on clone systems separately from the Solaris operating environment, the Solaris package database has no record of the additional software.

Creating Web Start Flash Archives (Tasks)

This chapter provides the procedures for creating a Web Start Flash archive.

- “Task Map: Creating Web Start Flash Archives” on page 113
- “Creating Web Start Flash Archives Tasks” on page 113

Task Map: Creating Web Start Flash Archives

TABLE 18-1 Task Map: Creating a Web Start Flash Archive

Task	Description	For Instructions
Install your chosen configuration on the master system.	Determine the configuration that meets your needs and use any of the Solaris installation methods to install the master system.	Chapter 2
Create the Web Start Flash archive.	Use the <code>flarcreate</code> command to create an archive.	“To Create a Web Start Flash Archive” on page 114

Creating Web Start Flash Archives Tasks

This section provides the procedures for installing a master system and then creating a Web Start Flash archive from that master system.

▼ To Install the Master System

You install the master system with the configuration that you want other systems to have. Use any of the Solaris installation methods to install the Solaris operating environment on the master system.

1. **Identify the system configuration that you want to install.**
2. **With the use of the Solaris installation methods, install the Solaris operating environment on the master system. For a discussion of the different installation methods, refer to Chapter 3.**
3. **Customize your Solaris installation in any of the following ways:**
 - Delete software
 - Add software
 - Modify configuration files
 - Add support for peripheral devices on the clone system

▼ To Create a Web Start Flash Archive

After you install the master system, create a Web Start Flash archive to use to install other systems.

1. **Boot the master system and run it in as inactive a state as possible.**

When possible, run the system in single-user mode. If that is not possible, shut down any applications that you want to archive and any applications that require extensive operating system resources.

You can create a Web Start Flash archive while the master system is running in multiuser mode, single-user mode, or while booted from one of the following:

 - Solaris 9 DVD.
 - Solaris 9 Software 1 of 2 CD.
 - An image of the Solaris 9 Software. If you are using CD media, the image could include the Solaris 9 Languages CD if needed.
2. **To create the archive, use the `flarcreate` command.**

```
# flarcreate -n name options path/filename
```

name The name that you give the archive. The *name* you specify is the value of the `content_name` keyword.

path The path to the directory in which you want to save the archive file. If you do not specify a path to save the archive, `flarcreate` saves the archive file in the current directory.

filename The name of the archive file.

For a list of command-line options, refer to “flarcreate” on page 126.

If the archive creation is successful, the `flarcreate` command returns an exit code of 0. If the archive creation fails, the `flarcreate` command returns a nonzero exit code.

Installing Web Start Flash Archives (Tasks)

This chapter provides the procedures for installing Web Start Flash archives on clone systems. You can use any of the Solaris installation methods to install Web Start Flash archives.

- “Installing Web Start Flash Archives With Solaris Live Upgrade” on page 117
- “Installing Web Start Flash Archives With the Solaris Web Start Program” on page 117
- “Installing Web Start Flash Archives With the Solaris `suninstall` Program” on page 119
- “Installing Web Start Flash Archives With a Custom JumpStart Installation” on page 120

Installing Web Start Flash Archives With Solaris Live Upgrade

For procedures on using Solaris Live Upgrade to install Web Start Flash archives, see “Installing Web Start Flash Archives on a Boot Environment” on page 278.

Installing Web Start Flash Archives With the Solaris Web Start Program

This section provides the procedure for using the Solaris Web Start program on the Solaris 9 DVD or Solaris 9 Installation CD to install Web Start Flash archives that are stored on the following:

- Disc (DVD or CD)
- NFS server
- HTTP server
- FTP server
- Local tape

▼ To Install a Web Start Flash Archive With the Solaris Web Start Program

1. **Begin the Solaris Web Start installation as described in Chapter 14.**
2. **On the Specify Media panel, select the location of the Web Start Flash archive.**
The Solaris Web Start program prompts you to proceed, depending on the media you selected.
3. **Type the information that you are prompted to enter.**

Media Selected	Prompt
DVD or CD	Insert the disc where the Web Start Flash archive is located.
Network File System	Specify the path to the network file system where the Web Start Flash archive is located. You can also specify the archive file name.
HTTP	Specify the URL and proxy information that is needed to access the Web Start Flash archive.
FTP	Specify the FTP server and the path to the Web Start Flash archive. Specify the user and password information that allows you to access to the FTP server. Specify any proxy information that is needed to access the FTP server.
Local tape	Specify the local tape device and the position on the tape where the Web Start Flash archive is located.

If you selected to install an archive from a DVD, CD, or from an NFS server, the Select Flash Archives panel is displayed.

4. **For archives that are stored on a disc or an NFS server, on the Select Flash Archives panel, select one or more Web Start Flash archives to install.**
5. **On the Flash Archives Summary panel, confirm the selected archives and click Next.**
6. **On the Additional Flash Archives panel, you can select to install layered Web Start Flash archives by specifying the media where another archive is located. If you do not want to install additional archives, select None and click Next to continue the installation.**

Installing Web Start Flash Archives With the Solaris `suninstall` Program

This section provides the procedure for using the Solaris `suninstall` program on the Solaris 9 Software 1 of 2 CD to install Web Start Flash archives that are stored on the following:

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

▼ To Install a Web Start Flash Archive With the Solaris `suninstall` Program

1. **Begin the Solaris `suninstall` program as described in Chapter 15.**
2. **On the Flash Archive Retrieval Method screen, select the location of the Web Start Flash archive.**

The Solaris `suninstall` program installation prompts you to proceed, depending on the media you selected.

3. **Type the information that you are prompted to enter.**

Media Selected	Prompt
HTTP	Specify the URL and proxy information that is needed to access the Web Start Flash archive.
FTP	Specify the FTP server and the path to the Web Start Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.
Network File System	Specify the path to the network file system where the Web Start Flash archive is located. You can also specify the archive file name.
Local file	Specify the path to the local file system where the Web Start Flash archive is located.

Media Selected	Prompt
Local tape	Specify the local tape device and the position on the tape where the Web Start Flash archive is located.
Local device	Specify the local device, the path to the Web Start Flash archive, and the type of file system on which the Web Start Flash archive is located.

4. On the Flash Archive Selection screen, you can select to install layered Web Start Flash archives by selecting New. If you do not want to install additional archives, press Continue to complete the installation.

Installing Web Start Flash Archives With a Custom JumpStart Installation

This section provides the procedure for using the custom JumpStart installation method to install Web Start Flash archives that are stored on the following:

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

▼ To Install a Web Start Flash Archive With a Custom JumpStart Installation

1. On the install server, create the custom JumpStart rules file.
For detailed instructions about creating custom JumpStart files, refer to Chapter 23.
2. On the install server, create the custom JumpStart profile file.
 - a. Set the value of the keyword `install_type` as `flash_install`.
 - b. Add the path to the Web Start Flash archive by using the new `archive_location` keyword.
Refer to “archive_location Keyword” on page 196, for details about the `archive_location` keyword.
 - c. Specify the file system configuration.

The Web Start Flash archive extraction process does not support auto-layout of partitions.

- d. **(Optional) If you want to install layered Web Start Flash archives on the clone system, add one `archive_location` line for each archive that you want to install.**

Note – From the existing list of custom JumpStart keywords in Chapter 23, the only keywords valid when you install a Web Start Flash archive are the following:

- `filesystem` – You cannot set the `filesystem` keyword to the value `auto`.
 - `install_type` (required).
 - `partitioning` – You can only set the `partitioning` keyword to the values `explicit` or `existing`.
-

3. **On the install server, add the clients that you are installing with the Web Start Flash archive.**

For detailed instructions, refer to “Adding Systems to Be Installed From the Network” on page 87.

4. **Perform the custom JumpStart installation on the clone systems.**

For detailed instructions, refer to Chapter 26.

Web Start Flash Custom JumpStart Profile Examples

The following are examples of profiles that can be used to install a Web Start Flash archive with the custom JumpStart installation method.

In the following example, the profile indicates that the custom JumpStart program retrieves the Web Start Flash archive from an HTTP server.

EXAMPLE 19–1 Installing a Web Start Flash Archive From an HTTP Server

```
install_type flash_install
archive_location http installserver /flasharchive/solaris9archive
partitioning explicit
filesystem c0t1d0s0 4000 /
filesystem c0t1d0s1 512 swap
filesystem c0t1d0s7 free /export/home
```

In the following example, the profile indicates that the custom JumpStart program retrieves the Web Start Flash archive from an NFS server.

EXAMPLE 19-2 Installing a Web Start Flash Archive From an NFS Server

```
install_type flash_install
archive_location nfs installserver:/export/solaris/flasharchive/solaris9archive
partitioning explicit
fileys rootdisk.s0 6000 /
fileys rootdisk.s1 512 swap
fileys rootdisk.s7 free /export/home
```

Web Start Flash (Reference)

The Web Start Flash installation feature provides tools for administering and managing Web Start Flash archives. After you create an archive, you can extract archive information, split an archive into sections, or combine archive sections.

- “Web Start Flash Archive Sections” on page 123
- “Web Start Flash Keywords” on page 124
- “Web Start Flash Commands” on page 126

Web Start Flash Archive Sections

Web Start Flash archives contain at least three sections. The sections include both archive identification information and the actual files that were copied from the master system to be installed on the Clone system.

1. Archive Cookie Section – The first section of a Web Start Flash archive contains a `cookie` that identifies the file as a Web Start Flash archive. The `cookie` must be present for an archive to be valid.
2. Archive Identification Section – The second section contains keywords with values that provide identification information about the archive.
3. User-Defined Sections – Following the Archive Identification Section, you can define and insert sections. The Web Start Flash archive does not process any sections that you insert. User-defined sections must be line oriented and terminated with newline (ASCII 0x0a) characters. The length of individual lines has no limit. If you include binary data in a user-defined section, you must encode it by using base64 or a similar algorithm.

The name for user-defined sections must begin with “X” and can contain any characters other than linefeeds, equal signs, null characters, and forward slashes (/). For example, “X-department” is a valid user-defined section name.

4. Archive Files Section – The Archive Files Section contains the files that were saved from the master system.

Web Start Flash Keywords

Keywords and values are separated by a single equal sign with only one pair per line. Individual lines can be any length. The keywords are case insensitive.

General Keywords

Each Web Start Flash archive section is defined by the `section_begin` and `section_end` keywords. The values for the `section_begin` and `section_end` keywords are described in the following table.

TABLE 20-1 Values for `section_begin` and `section_end` Keywords

Archive Section	Value for <code>section_begin</code> and <code>section_end</code> keywords
Archive cookie	<code>cookie</code>
Archive identification	<code>identification</code>
User-defined sections	<code>section_name</code>
Archive files	<code>archive</code>

Identification Section Keywords

This section describes the keywords for use in the Archive Identification section and the values you can define for them.

The following table explains keywords that describe the archive.

TABLE 20–2 Identification Section Keywords

Keywords	Value Definitions
<code>content_name</code> (required)	<p>The Web Start Flash archive deployment utilities use the value of the <code>content_name</code> keyword to identify the archive. The value can be no longer than 256 characters.</p> <p>You might want the value of the <code>content_name</code> keyword to describe the function and purpose of the archive because the <code>content_name</code> value might be presented to the user during the archive selection and extraction processes.</p>
<code>creation_date</code>	<p>The value of the <code>creation_date</code> keyword is a textual timestamp that represents the time that you created the archive. The value must be in the format <code>YYYYMMDDhhmmss</code>. For example, <code>20000131221409</code> represents January 31st, 2000 10:14:09p.m. If you do not specify a creation date, the default date is set in Greenwich mean time (GMT).</p>
<code>creation_master</code>	<p>The value of the <code>creation_master</code> keyword is the name of the master system you used to create the archive. If you do not specify a value for <code>creation_master</code>, <code>flarcreate</code> uses the system name that is reported by <code>uname -n</code>.</p>
<code>content_type</code>	<p>You define the value of the <code>content_type</code> keyword to specify a category for the archive. The Web Start Flash archive deployment utilities display the value of the <code>content_type</code> keyword during deployment.</p>
<code>content_description</code>	<p>You define the value of the <code>content_description</code> keyword to provide a description of the contents of the archive. The value of this keyword has no length limit.</p>
<code>content_author</code>	<p>You define the value of the <code>content_author</code> keyword to identify the creator of the archive. Suggested values include the full name of the creator and the creator's email address.</p>
<code>content_architectures</code>	<p>The value of the <code>content_architectures</code> keyword is a comma-delimited list of the kernel architectures that the archive supports. When you create a Web Start Flash archive, the archive generates the value of the <code>content_architectures</code> keyword.</p> <p>If the archive contains this keyword, the Web Start Flash archive deployment utilities validate the kernel architecture of the clone system against the list of architectures that the archive supports. The deployment fails if the archive does not support the kernel architecture of the clone system. If the keyword is not present, the deployment utilities do not validate the architecture of the clone system.</p>
<code>creation_node</code>	<p>The value returned from <code>uname -n</code>. If you create a Web Start Flash archive in which the root directory is not <code>/</code>, <code>flarcreate</code> uses the contents of the <code>nodename(4)</code> file.</p>
<code>creation_hardware_class</code>	<p>The value returned from <code>uname -m</code>. If you create a Web Start Flash archive in which the root directory is not <code>/</code>, <code>flarcreate</code> inserts the string <code>UNKNOWN</code>.</p>

TABLE 20-2 Identification Section Keywords (Continued)

Keywords	Value Definitions
<code>creation_platform</code>	The value returned from <code>uname -i</code> . If you create a Web Start Flash archive in which the root directory is not <code>/</code> , <code>flarcreate</code> inserts the string UNKNOWN.
<code>creation_processor</code>	The value returned from <code>uname -p</code> . If you create a Web Start Flash archive in which the root directory is not <code>/</code> , <code>flarcreate</code> inserts the string UNKNOWN.
<code>creation_release</code>	The value returned from <code>uname -r</code> . If you create a Web Start Flash archive in which the root directory is not <code>/</code> , <code>flarcreate</code> attempts to use the contents of <code>root_directory/var/sadm/system/admin/INST_RELEASE</code> . If <code>flarcreate</code> cannot read this file, it inserts the string UNKNOWN.
<code>creation_os_name</code>	The value returned from <code>uname -s</code> . If you create a Web Start Flash archive in which the root directory is not <code>/</code> , <code>flarcreate</code> attempts to use the contents of <code>root_directory/var/sadm/system/admin/INST_RELEASE</code> . If <code>flarcreate</code> cannot read this file, it inserts the string UNKNOWN.
<code>creation_os_version</code>	The value returned from <code>uname -v</code> . If you create a Web Start Flash archive in which the root directory is not <code>/</code> , <code>flarcreate</code> inserts the string UNKNOWN.

In addition to the keywords that are defined by the Web Start Flash archive, you can define keywords. The Web Start Flash archive ignores user-defined keywords, but you can provide scripts or programs that process the identification section and use user-defined keywords. The name of a user-defined keyword must begin with “X” and can contain any characters other than linefeeds, equal signs, and null characters. For example, X-department is a valid name for a user-defined keyword.

Web Start Flash Commands

Use the Web Start Flash commands to create and manage Web Start Flash archives.

`flarcreate`

Use the `flarcreate` command to create a Web Start Flash archive from a master system. You can use this command when the master system is running in multiuser mode or single-user mode. You can also use `flarcreate` when the master system is booted from the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD or from an image of the Solaris 9 Software and Solaris 9 Languages CDs. The master system should be in as stable a state as possible when you create a Web Start Flash archive. The syntax of the command is as follows:

```
flarcreate -n name [-R root] [-S] [-H] [-c] [-x exclude] [-t [-p
posn] [-b blocksize]] [-i date] [-m [-u section [-d dir]] [-f file_list] [-F]
[-U key=val] master] [-a author] [-e descr:-E descr_file] [-T type] path/filename
```

In this command line, *path* is the directory in which you want the archive file to be saved and *filename* is the name of the archive file. If you do not specify a path, `flarcreate` saves the archive file in the current directory.

TABLE 20-3 Command-Line Options for `flarcreate`

Option	Description
Required Options	
-n <i>name</i>	The value of this flag is the name of the archive. The <i>name</i> you specify is the value of the <code>content_name</code> keyword.
Option for Compression	
-c	Compresses the archive by using <code>compress(1)</code> .
Options for Directories and Sizes	
-R <i>root</i>	Creates the archive from the file system tree that is rooted at <i>root</i> . If you do not specify this option, <code>flarcreate</code> creates an archive from a file system that is rooted at <code>/</code> .
-S	Does not include sizing information in the archive.
-H	Does not generate the hash identifier.
Option for Excluding Files and Directories	
-x <i>exclude</i>	Excludes the file or directory <i>exclude</i> from the archive. If you specify a file system with -R <i>root</i> , the path to the directory <i>exclude</i> is assumed to be relative to <i>root</i> .
Options Used With User-Defined Sections	
-u <i>section</i>	Includes a user-defined section. To include more than one user-defined section, <i>section</i> must be a space-separated list of section names.
-d <i>dir</i>	Retrieves the section file that is specified with -u from <i>dir</i> .
Options Used With Tape Archives	
-t	Creates an archive on a tape device. The <i>filename</i> argument is the name of the tape device.
-p <i>posn</i>	Use only with the -t option. Specifies the position on the tape device for <code>flarcreate</code> to store the archive. If you do not use this option, <code>flarcreate</code> places the archive in the current position of the tape.

TABLE 20-3 Command-Line Options for `flarcreate` (Continued)

Option	Description
<code>-b <i>blocksize</i></code>	Specifies the block size <code>flarcreate</code> uses when creating the archive. If you do not specify a block size, <code>flarcreate</code> uses the default block size of 64k.
Options for Specifying Files	
<code>-f <i>file_list</i></code>	Adds the files in the file <i>file_list</i> to the archive. The <i>file_list</i> file must contain one file per line. The path to each file must be relative to the alternate root directory or an absolute path. If you use "-" as the value of <i>file_list</i> , <code>flarcreate</code> uses the output of <code>stdin</code> as the list of files. When you use the value "-", the archive size is not calculated.
<code>-F</code>	Uses only the files in <i>file_list</i> to create the archive.
Options for Archive Identification	
<code>-U <i>key=val</i></code>	Includes user-defined keyword(s) and values in the Archive Identification section.
<code>-i <i>date</i></code>	Uses <i>date</i> as the value for the <code>creation_date</code> keyword. If you do not specify a date, <code>flarcreate</code> uses the current system time and date.
<code>-m <i>master</i></code>	Uses <i>master</i> as the name of the master system on which you created the archive for the <code>creation_master</code> keyword. If you do not specify a <i>master</i> , <code>flarcreate</code> uses the system name that is reported by <code>uname -n</code> .
<code>-e <i>descr</i></code>	Uses <i>descr</i> for the value of the <code>content_description</code> keyword. You cannot use this option when you use the <code>-E</code> option.
<code>-E <i>descr_file</i></code>	Retrieves the value for the <code>content_description</code> keyword from the file <i>descr_file</i> . You cannot use this option when you use the <code>-e</code> option.
<code>-a <i>author</i></code>	Uses <i>author</i> as the author name in the Archive Identification section for the <code>content_author</code> keyword. If you do not specify an author, <code>flarcreate</code> does not include the <code>content_author</code> keyword in the Archive Identification section.
<code>-T <i>type</i></code>	Uses <i>type</i> as the value for the <code>content_type</code> keyword. If you do not specify a type, <code>flarcreate</code> does not include the <code>content_type</code> keyword.

flar

The `flar` command allows you to administer archives. With the `flar` command you can accomplish the following tasks.

- “Extracting Information From an Archive” on page 129
- “Splitting Archives” on page 129
- “Combining Archives” on page 130

Extracting Information From an Archive

Use the `flar` command with the `-i` option to get information about archives you have already created. The syntax of the command is as follows:

```
flar -i:info [-l] [-k keyword] [-t [-p posn] [-b blocksize]] filename
```

TABLE 20-4 Command-Line Options for `flar -i`

Option	Description
<code>-k keyword</code>	Returns only the value of the keyword <i>keyword</i> .
<code>-l</code>	Lists all the files in the archive section.

Splitting Archives

The `flar` command with the `-s` option splits a Web Start Flash archive into sections. The `flar` command copies each section into a separate file in the current or specified directory. The files are named after the sections, for example, the archive cookie is saved in a file that is named `cookie`. You can specify that the `flar` command only save one section. The syntax of the command is as follows:

```
flar -s:split [-d dir] [-u section] [-f archive] [-S section] [-t [-p posn] [-b blocksize]] filename
```

TABLE 20-5 Command-Line Options for `flar -s`

Option	Description
<code>-d dir</code>	Retrieves the sections to copy from <i>dir</i> , rather than from the current directory.
<code>-u section</code>	If you do not use this option, <code>flar</code> copies all sections in the current directory. If you use this option, <code>flar</code> copies the Cookie, Identification, Archive, and <i>section</i> sections. You can specify a single section name or a space-separated list of section names.
<code>-f archive</code>	Extracts the Archive section into a directory that is named <i>archive</i> , rather than places it in a file with the name <i>archive</i> .
<code>-S section</code>	Only copies the section that is named <i>section</i> from the archive.

Combining Archives

The `flar` command with the `-c` option creates a Web Start Flash archive from individual sections. Each section is assumed to be in a separate file, the names of which are the section names. At a minimum, the Archive Cookie (`cookie`), Archive Identification (`identification`), and Archive Files (`archive`) sections must be present. If `archive` is a directory, the `flar` command uses `cpio` to archive it before including it in the combined archive. If the Archive Identification section specifies to compress the archive, `flar` compresses the contents of the newly combined archive.

```
flar -c:combine [-d dir] [-u section] [-t [-p posn] [-b blocksize]] filename
```

Note – No validation is performed on any of the sections. In particular, no fields in the Archive Identification section are validated or updated.

TABLE 20-6 Command-Line Options for `flar -c`

Option	Description
<code>-d <i>dir</i></code>	Retrieves the sections to combine from <i>dir</i> , rather than from the current directory.
<code>-u <i>section</i></code>	If you do not use this option, <code>flar</code> combines all sections in the current directory. If you use this option, <code>flar</code> combines only the Cookie, Identification, Archive, and <i>section</i> sections. You can specify a single section name or a space-separated list of section names.

Custom JumpStart Installation (Topics)

This section provides instructions for creating, preparing, and performing custom JumpStart installations.

Chapter 22	Provides an introduction and overview of the custom JumpStart installation method.
Chapter 23	Provides instructions about how to prepare the systems at your site from which and on which you are installing the Solaris 9 software with the custom JumpStart installation method.
Chapter 24	Describes the optional features that you can use to create additional tools for a custom JumpStart installation.
Chapter 25	Provides information and procedures for creating your own custom rule and custom probe keywords.
Chapter 26	Describes how to perform a custom JumpStart installation. You need to follow these procedures on the system on which you intend to install the Solaris 9 software.
Chapter 27	Provides an example of setting up and installing Solaris software by using the custom JumpStart installation method.
Chapter 28	Contains lists of keywords and values to be used in the <code>rules</code> file, profiles, <code>begin</code> scripts, and <code>finish</code> scripts.

Custom JumpStart (Overview)

This chapter provides an introduction and overview to the custom JumpStart installation process.

- “Custom JumpStart Introduction” on page 133
- “How the JumpStart Program Installs Solaris Software” on page 134

Custom JumpStart Introduction

The custom JumpStart installation method is a command-line interface that enables you to automatically install or upgrade several systems, based on profiles that you create. The profiles define specific software installation requirements. You can also incorporate shell scripts to include preinstallation and postinstallation tasks. You choose which profile and scripts to use for installation or upgrade. The custom JumpStart installation method installs or upgrades the system, based on the profile and scripts that you select. Also, you can use a `sysidcfg` file to specify configuration information so that the custom JumpStart installation is completely hands-off.

The custom JumpStart process can be described by using an example scenario. In this example scenario, the systems need to be set up with the following parameters:

- Install Solaris on 100 new systems.
- Seventy of the systems are owned by the engineering group and need to be installed as standalone systems with the Solaris operating environment software group for developers.
- The remaining 30 systems are owned by the marketing group and need to be installed as standalone systems with the Solaris operating environment software group for end users.

First, the system administrator must create a `rules` file and a profile for each group of systems. The `rules` file is a text file that contains a rule for each group of systems or single systems on which you want to install the Solaris software. Each rule distinguishes a group of systems that are based on one or more system attributes. Each rule also links each group to a profile.

A profile is a text file that defines how the Solaris software is to be installed on each system in the group. Both the `rules` file and profile must be located in a JumpStart directory.

For the example scenario, the system administrator creates a `rules` file that contains two different rules, one for the engineering group and another for the marketing group. For each rule, the system's network number is used to distinguish the engineering group from the marketing group.

Each rule also contains a link to an appropriate profile. For example, in the rule for the engineering group, a link is added to the profile, `eng_profile`, which was created for the engineering group. In the rule for the marketing group, a link is added to the profile, `market_profile`, which was created for the marketing group.

You can save the `rules` file and the profiles on a diskette or on a server.

- A profile diskette is required when you want to perform custom JumpStart installations on non-networked, standalone systems.
- A profile server is used when you want to perform custom JumpStart installations on networked systems that have access to a server.

After creating the `rules` file and profiles, validate the files with the `check` script. If the `check` script runs successfully, the `rules.ok` file is created. The `rules.ok` is a generated version of the `rules` file that the JumpStart program uses to install the Solaris software.

How the JumpStart Program Installs Solaris Software

After you validate the `rules` file and the profiles, you can begin a custom JumpStart installation. The JumpStart program reads the `rules.ok` file. Then, the JumpStart program searches for the first rule with defined system attributes that match the system on which the JumpStart program is attempting to install the Solaris software. If a match occurs, the JumpStart program uses the profile that is specified in the rule to install the Solaris software on the system.

Figure 22-1 illustrates how a custom JumpStart installation works on a standalone, non-networked system. The system administrator initiates the custom JumpStart installation on Pete's system. The JumpStart program accesses the rules files on the diskette in the system's diskette drive. The JumpStart program matches rule 2 to the system. rule 2 specifies that the JumpStart program use Pete's profile to install the Solaris software. The JumpStart program reads Pete's profile and installs the Solaris software, based on the instructions that the system administrator specified in Pete's profile.

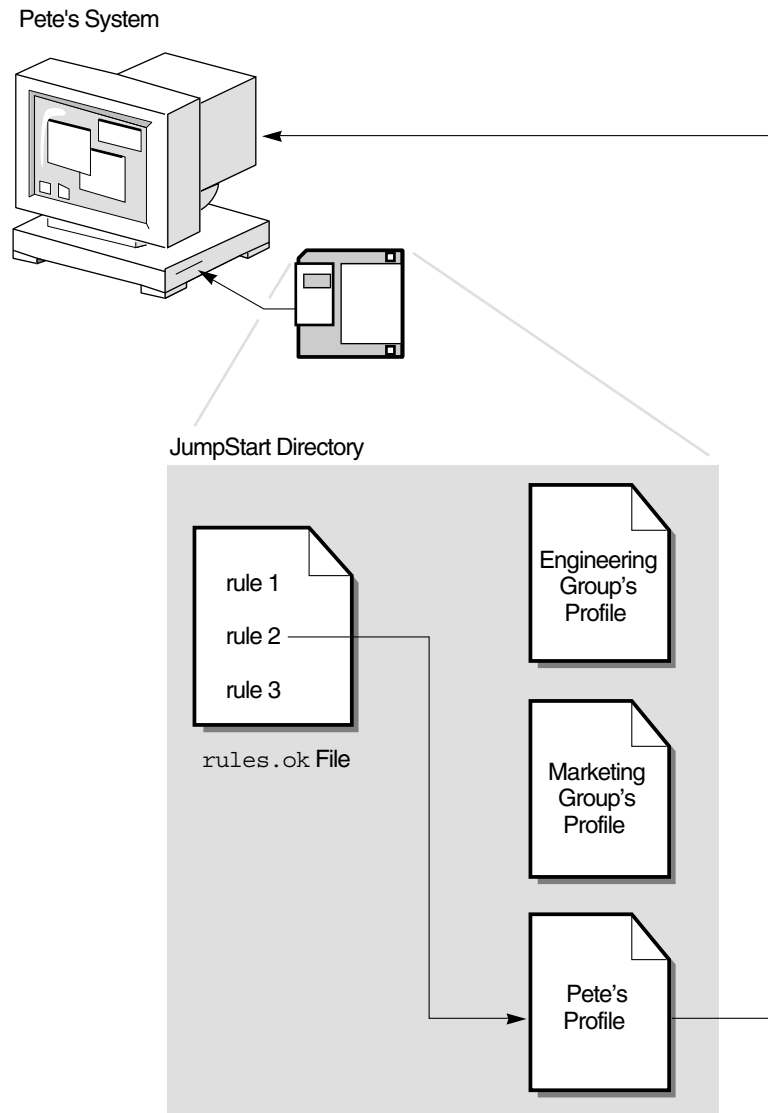


FIGURE 22-1 How a Custom JumpStart Installation Works: Non-Networked Example

Figure 22-2 illustrates how a custom JumpStart installation works with more than one system on a network. The system administrator set up different profiles and saved the profiles on a single server. The system administrator initiates the custom JumpStart installation on one of the engineering systems. The JumpStart program accesses the rules files in the JumpStart/ directory on the server. The JumpStart program matches the engineering system to rule 1. rule 1 specifies that the JumpStart

program use Engineering Group's Profile to install the Solaris software. The JumpStart program reads Engineering Group's Profile and installs the Solaris software, based on the instructions that the system administrator specified in Engineering Group's Profile.

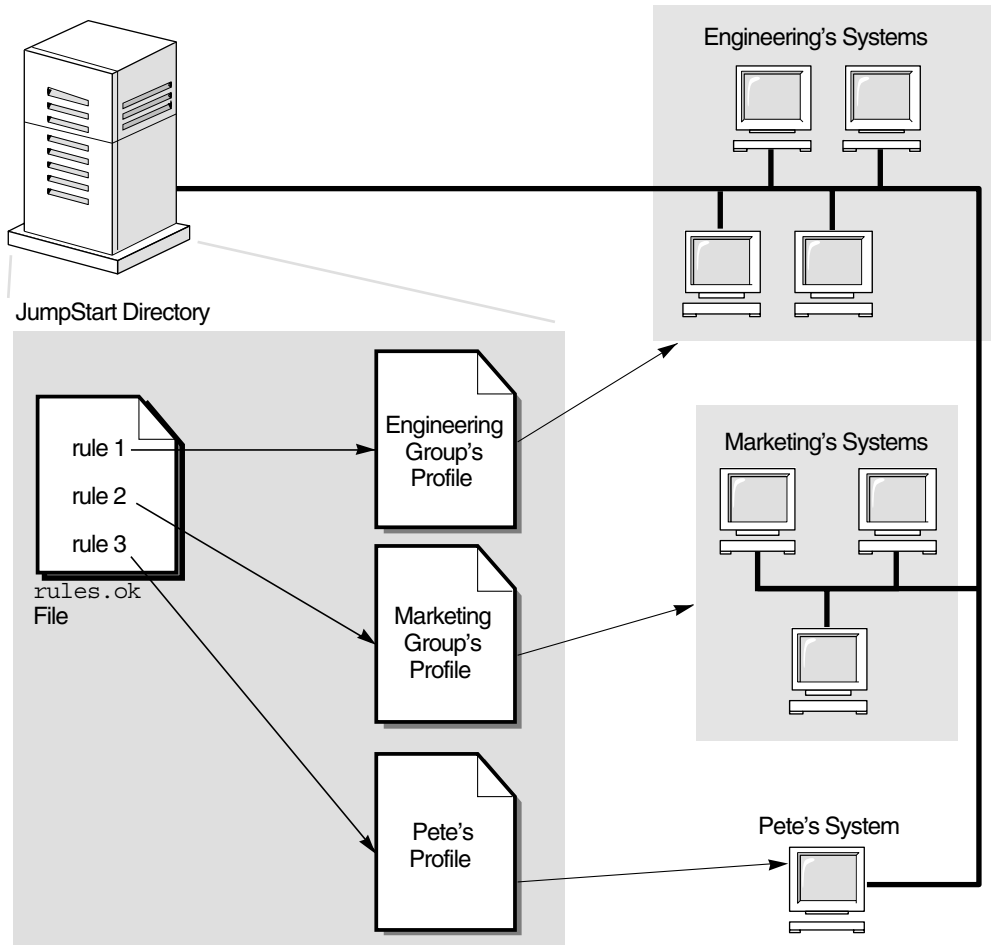


FIGURE 22-2 How a Custom JumpStart Installation Works: Networked Example

Figure 22-3 describes the order in which the JumpStart program searches for custom JumpStart files.

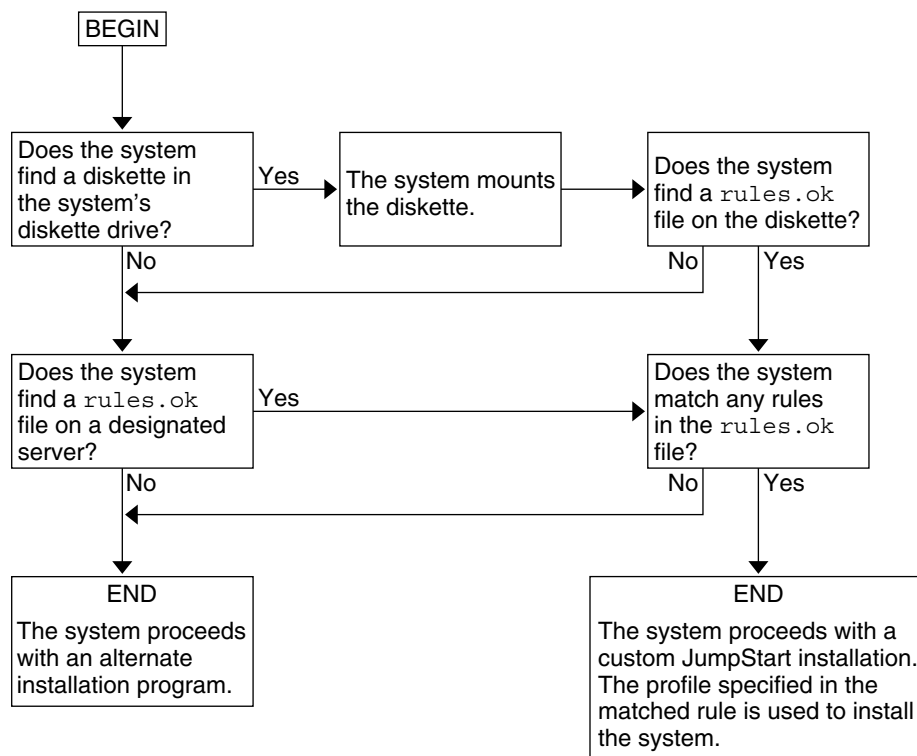


FIGURE 22-3 What Happens During a Custom JumpStart Installation

Preparing Custom JumpStart Installations (Tasks)

This chapter provides step-by-step instructions about how to prepare the systems at your site from which and on which you intend to install the Solaris 9 software by using the custom JumpStart installation method.

- “Task Map: Preparing Custom JumpStart Installations” on page 139
- “Creating a Profile Server for Networked Systems” on page 141
- “Creating a Profile Diskette for Standalone Systems” on page 144
- “Creating the rules File” on page 147
- “Creating a Profile” on page 150
- “Testing a Profile” on page 153
- “Validating the rules File” on page 157

Task Map: Preparing Custom JumpStart Installations

TABLE 23-1 Task Map: Preparing Custom JumpStart Installations

Task	Description	For Instructions
Decide how to upgrade the system if a previous version of the Solaris software is installed on the system	If a previous release of Solaris is installed on the system, you need to determine how to upgrade the system. Ensure that you know what to do before and after you upgrade a system. Planning helps you to create your profiles, begin scripts, and finish scripts.	Chapter 8

TABLE 23-1 Task Map: Preparing Custom JumpStart Installations (Continued)

Task	Description	For Instructions
Create a JumpStart directory	<p>On a server</p> <p>If you want to perform custom JumpStart installations on systems that are connected to a network, you must create a profile server. The profile server contains a JumpStart directory for the custom JumpStart files.</p> <p>On a diskette</p> <p>If you want to perform custom JumpStart installations on systems that are not connected to a network, you must create a profile diskette. A profile diskette contains the custom JumpStart files.</p>	<p>“Creating a Profile Server for Networked Systems” on page 141</p> <p>“Creating a Profile Diskette for Standalone Systems” on page 144</p>
Add rules to the rules file	<p>After you decide how you want each group of systems or single systems to be installed, create a rule for each group that you want to install. Each rule distinguishes a group, based on one or more system attributes. The rule links each group to a profile.</p>	<p>“Creating the rules File” on page 147</p>
Create a profile for every rule	<p>A profile is a text file that defines how to install the Solaris software, for example, which software group to install on a system. Every rule specifies a profile to define how a system is to be installed with the Solaris software when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule.</p>	<p>“Creating a Profile” on page 150</p>
(Optional) Test the profiles	<p>After you create a profile, use the <code>pfinstall(1M)</code> command to test the profile before you use the profile to install or upgrade a system.</p>	<p>“Testing a Profile” on page 153</p>

TABLE 23-1 Task Map: Preparing Custom JumpStart Installations (Continued)

Task	Description	For Instructions
Validate the <code>rules</code> file	The <code>rules.ok</code> file is a generated version of the <code>rules</code> file that the JumpStart program uses to match the system to be installed with a profile. You must use the <code>check</code> script to validate the <code>rules</code> file.	"Validating the <code>rules</code> File" on page 157

Creating a Profile Server for Networked Systems

When setting up custom JumpStart installations for systems on the network, you need to create a directory on a server that is called a JumpStart directory. The JumpStart directory contains all of the essential custom JumpStart files, for example, the `rules` file, `rules.ok` file, and profiles. You must save the JumpStart directory in the root (`/`) directory of the profile server.

The server that contains a JumpStart directory is called a profile server. A profile server can be the same system as an install server or a boot server, or the server can be a completely different server.

Note – After you create a profile server, you must allow systems to access the server. For detailed instructions, see "To Allow All Systems Access to the Profile Server" on page 143.

▼ To Create a JumpStart Directory on a Server

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

1. Log in as superuser on the server on which you want to create the JumpStart directory.
2. Create the JumpStart directory anywhere on the server.

```
# mkdir -m 755 jumpstart_dir_path
```

In the command, *jumpstart_dir_path* is the absolute path of the JumpStart directory.

For example, the following command creates a directory that is called `jumpstart` in the root (/) directory and sets the permissions to `755`:

```
# mkdir -m 755 /jumpstart
```

3. Edit the /etc/dfs/dfstab file by adding the following entry.

```
share -F nfs -o ro,anon=0 jumpstart_dir_path
```

For example, the following entry shares the /jumpstart directory:

```
share -F nfs -o ro,anon=0 /jumpstart
```

4. Type `shareall` and press Enter.

5. Determine if you want to copy examples of custom JumpStart files to your JumpStart directory.

- If no, go to Step 8.
- If yes, use the following decision table to determine what to do next.

Example Locations	Instructions
The Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD for your platform	Insert the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD into the server's CD-ROM drive. Volume Manager automatically mounts the CD.
An image of the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD for your platform on a local disk	Change directory to the location of the Solaris 9 DVD or the Solaris 9 Software 1 of 2 image. For example, type the following command: <code>cd /export/install</code>

6. Copy the example custom JumpStart files into the JumpStart directory on the profile server.

```
# cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
```

<i>media_path</i>	The path to the CD, DVD, or image on the local disk
<i>jumpstart_dir_path</i>	The path on the profile server where you are placing the example custom JumpStart files

For example, the following command copies the `jumpstart_sample` directory into the /jumpstart directory on the profile server:

```
cp -r /cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample/* /jumpstart
```

7. Update the example JumpStart files so that the files work in your environment.
8. Ensure that `root` owns the JumpStart directory and that the permissions are set to 755.
9. Allow systems on the network to access the profile server.
For detailed instructions, see “To Allow All Systems Access to the Profile Server” on page 143.

▼ To Allow All Systems Access to the Profile Server

When you create a profile server, you must ensure that systems can access the JumpStart directory on the profile server during a custom JumpStart installation. Use one of the following ways to ensure access:

- `add_install_client` command – Each time that you add a system for network installation, use the `-c` option with the `add_install_client` command. For detailed instructions, refer to “Adding Systems to Be Installed From the Network” on page 87.
- `boot` command – Specify the location of the JumpStart directory on the profile server when you boot the system. You must compress the custom JumpStart configuration files into one file. Then, save the compressed configuration file on an NFS server, an HTTP server, or on media that the system can access locally. For detailed instructions, refer to “Creating a Compressed Configuration File” on page 166.

When you boot the system to initiate the custom JumpStart installation, specify the location of the compressed file. For detailed instructions, refer to Step 5 in “To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 178.

- `/etc/bootparams` file – Use a wildcard in the `/etc/bootparams` file. Use the following steps to add a wildcard in the `etc/bootparams` file.

Note – The following procedure is not necessary if you save the JumpStart directory on a diskette or if you specify the location of the profile server when you boot the system.

The following procedure is valid only if you store network installation information in the `/etc/bootparams` file. You can also store network installation information in one of the following places:

- Name service database – If you store network installation information in the name service `bootparams` database, you must update the `bootparams` database with the entry that is shown in Step 3.

- DHCP server – If you store network installation information on a DHCP server, use the `boot` command to specify that the custom JumpStart program use the DHCP server. For detailed instructions, refer to Step 5 in “To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 178.

1. On the install or boot server, log in as superuser.
2. Use a text editor to open `/etc/bootparams`.
3. Add this entry.

```
* install_config=server:jumpstart_dir_path
```

*	A wildcard character that specifies that all systems have access
<i>server</i>	The host name of the profile server where the JumpStart directory is located
<i>jumpstart_dir_path</i>	The absolute path of the JumpStart directory

For example, the following entry allows all systems to access the `/jumpstart` directory on the profile server that is named `sherlock`:

```
* install_config=sherlock:/jumpstart
```



Caution – Use of this procedure might produce the following error message when an install client is booted:

```
WARNING: getfile: RPC failed: error 5: (RPC Timed out).
```

“Booting From the Network, Error Messages” on page 339 contains details about this error message.

All systems can now access the profile server.

Creating a Profile Diskette for Standalone Systems

A diskette that contains a JumpStart directory is called a profile diskette. A system that is not connected to the network does not have access to a profile server. As a result, you must create a JumpStart directory on a diskette if a system is not connected to a network. The system on which you create a profile diskette must have a diskette drive.

The JumpStart directory contains all of the essential custom JumpStart files, for example, the `rules` file, `rules.ok` file, and profiles. You must save the JumpStart directory in the root (`/`) directory of the profile diskette.

▼ To Create a Profile Diskette

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

1. Log in as superuser on a system to which a diskette drive is attached.
2. Insert a blank diskette or a diskette that can be overwritten in the diskette drive.
3. Mount the diskette.

```
# volcheck
```

4. Determine if the diskette contains a UNIX file system (UFS).

Examine the contents of the file `/etc/mnttab` on the system for an entry such as the following:

```
/vol/dev/diskette0/scrap /floppy/scrap ufs suid,rw,largefiles,dev=1740008 927147040
```

- If the entry exists, go to Step 7.
- If the entry does not exist, go to the next step.

5. Format the diskette.



Caution – Formatting erases all data on the diskette.

```
# fdformat -U
```

6. Create a UFS on the diskette.

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

7. Determine if you want to copy examples of custom JumpStart files to your JumpStart directory.

- If no, go to Step 10.
- If yes, use the following decision table to determine what to do next.

Example Locations	Instructions
The Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD	Insert the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD into the server's CD-ROM drive. Volume Manager automatically mounts the CD.
An image of the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD on a local disk	Change the directory to the location of the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD image. For example, type the following command: <code>cd /export/install</code>

8. Copy the example custom JumpStart files into the JumpStart directory on the profile diskette.

```
# cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
```

<i>media_path</i>	The path to the CD, DVD, or image on the local disk
<i>jumpstart_dir_path</i>	The path to the profile diskette where you want to place the example custom JumpStart files

Note – You must place all custom JumpStart installation files in the root (/) directory on the diskette.

For example, the following command copies the contents of `jumpstart_sample` on the Solaris 9 Software 1 of 2 CD to the root (/) directory on a profile diskette that is named `scrap`:

```
cp -r /cdrom/sol_9_sparc/s0/Solaris_9/Misc/jumpstart_sample/* /floppy/scrap
```

- 9. Update the example JumpStart files on the profile diskette so that the files work in your environment.**
- 10. Ensure that `root` owns the JumpStart directory and that permissions are set to 755.**
- 11. Eject the diskette.**

```
# eject floppy
```

You have completed the creation of a profile diskette. You can now update the rules file and create profiles on the profile diskette to perform custom JumpStart installations. To continue, go to “Creating the rules File” on page 147.

Creating the rules File

The `rules` file is a text file that contains a rule for each group of systems on which you want to install the Solaris operating environment. Each rule distinguishes a group of systems that are based on one or more system attributes. Each rule also links each group to a profile. A profile is a text file that defines how the Solaris software is to be installed on each system in the group. For example, the following rule specifies that the JumpStart program use the information in the `basic_prof` profile to install any system with the `sun4u` platform group.

```
karch sun4u - basic_prof -
```

The `rules` file is used to create the `rules.ok` file, which is required for custom JumpStart installations.

Note – If you set up the JumpStart directory by using the procedures in “Creating a Profile Diskette for Standalone Systems” on page 144 or “Creating a Profile Server for Networked Systems” on page 141, an example `rules` file is already located in the JumpStart directory. The sample `rules` file contains documentation and some example rules. If you use the sample `rules` file, ensure that you comment out the example rules you do not intend to use.

Syntax of the rules File

The `rules` file must have the following attributes:

- The file must be assigned the name `rules`.
- The file must contain at least one rule.

The `rules` file can contain any of the following:

- Commented text
Any text that is included after the `#` symbol on a line is treated by JumpStart as commented text. If a line begins with the `#` symbol, the entire line is treated as a comment.
- One or more blank lines
- One or more multiline rules
To continue a single rule onto a new line, include a backslash character (`\`) just before pressing Return.

▼ To Create a rules File

1. Use a text editor to create a text file that is named `rules`. Or, open the sample `rules` file in the JumpStart directory that you created.
2. Add a rule in the `rules` file for each group of systems on which you want to install the Solaris software.

For a list of `rules` file keywords and values, see “Rule Keywords and Values” on page 189.

A rule within a `rules` file must adhere to the following syntax:

```
[!]rule_keyword rule_value [&& [!]rule_keyword rule_value] ... begin profile finish
```

TABLE 23-2 Syntax Elements of a Rule

Element	Description
!	A symbol that is used before a keyword to indicate negation.
<i>rule_keyword</i>	A predefined lexical unit or word that describes a general system attribute, such as host name, <code>hostname</code> , or memory size, <code>memsize</code> . <i>rule_keyword</i> is used with the rule value to match a system with the same attribute to a profile. For the list of rule keywords, see “Rule Keywords and Values” on page 189.
<i>rule_value</i>	A value that provides the specific system attribute for the corresponding rule keyword. Rule values are described in “Rule Keywords and Values” on page 189.
&&	A symbol you must use to join rule keyword and rule value pairs in the same rule (a logical AND). During a custom JumpStart installation, a system must match every pair in the rule before the rule matches.
<i>begin</i>	The name of an optional Bourne shell script that can be executed before the installation begins. If no <i>begin</i> script exists, you must type a minus sign (-) in this field. All <i>begin</i> scripts must be located in the JumpStart directory. Information about how to create <i>begin</i> scripts is presented in “Creating Begin Scripts” on page 159.
<i>profile</i>	The name of a text file that defines how the Solaris software is to be installed on the system when a system matches the rule. The information in a profile consists of profile keywords and their corresponding profile values. All profiles must be located in the JumpStart directory. Note – Optional ways to use the <i>profile</i> field are described in “Using a Site-Specific Installation Program” on page 169 and “Creating Derived Profiles With a Begin Script” on page 160.

TABLE 23-2 Syntax Elements of a Rule (Continued)

Element	Description
<i>finish</i>	The name of an optional Bourne shell script that can be executed after the installation is completed. If no finish script exists, you must type a minus sign (-) in this field. All finish scripts must be located in the JumpStart directory. Information about how to create finish scripts is presented in “Creating Finish Scripts” on page 161.

At the minimum, each rule must contain the following:

- A keyword, a value, and a corresponding profile
- A minus sign (-) in the *begin* and *finish* fields if no begin or finish scripts are specified

3. Save the `rules` file in the JumpStart directory.
4. Ensure that `root` owns the `rules` file and that the permissions are set to 644.

rules File Example

The following example shows several example rules in a `rules` file. Each line has a rule keyword and a valid value for that keyword. The JumpStart program scans the `rules` file from top to bottom.

When the JumpStart program matches a rule keyword and value with a known system, the JumpStart program installs the Solaris software that is specified by the profile that is listed in the profile field.

EXAMPLE 23-1 rule File

```
# rule keywords and rule values      begin script      profile      finish script
# -----
hostname eng-11                    -                basic_prof   -
network 192.43.34.0 && !model \
'SUNW,SPARCstation-20'2            -                net_prof     -
model SUNW,SPARCstation-LX3        -                lx_prof      complete
network 193.144.2.0 && karch sun4u  setup4           4u_prof      done
memsize 64-128 && arch sparc5      -                prog_prof    -
any6 -                             -                generic_prof -
```

1. The rule matches if the system's host name is `eng-1`. The `basic_prof` profile is used to install the Solaris software on the system that matches the rule.
2. The rule matches if the system is on subnet 192.43.34.0 and if the system is *not* a SPARCstation™ 20 (SUNW, SPARCstation-20). The `net_prof` profile is used to install the Solaris software on systems that match this rule. The rule also provides an example of rule wrap, which is defined in “Syntax of the `rules` File”

EXAMPLE 23-1 rule File (Continued)

on page 147.

3. The rule matches if the system is a SPARCstation LX. The `lx_prof` profile and the `complete` finish script are used to install the Solaris software on systems that match this rule.
4. The rule matches if the system is on subnet 193.144.2.0 and is a sun4u system. The `setup` begin script, the `4u_prof` profile, and the `done` finish script are used to install the Solaris software on systems that match the rule.
5. The rule matches if the system has between 64 and 128 Mbytes of memory. The `prog_prof` profile is used to install the Solaris software on systems that match the rule.
6. The rule matches any system that did not match the previous rules. The `generic_prof` profile is used to install the Solaris software on systems that match the rule. If any is used, it should always be the last rule in the rules file.

Creating a Profile

A profile is a text file that defines how to install the Solaris software on a system. A profile defines elements of the installation, for example, the software group to install. Every rule specifies a profile that defines how a system is to be installed. You can create different profiles for every rule or the same profile can be used in more than one rule.

A profile consists of one or more profile keywords and their values. Each profile keyword is a command that controls one aspect of how the JumpStart program is to install the Solaris software on a system. For example, the following profile keyword and value specify that the JumpStart program install the system as a server:

```
system_type server
```

Note – If you created the JumpStart directory by using the procedures that are presented in “Creating a Profile Server for Networked Systems” on page 141 or “Creating a Profile Diskette for Standalone Systems” on page 144, sample profiles are already located in the JumpStart directory.

Syntax of Profiles

A profile must contain the following:

- The `install_type` profile keyword as the first entry

- One keyword per line
- The `root_device` keyword if the systems that are being upgraded by the profile contain more than one root (/) file system that can be upgraded

A profile can contain the following:

- Commented text
Any text that is included after the # symbol on a line is treated by the JumpStart program as commented text. If a line begins with the # symbol, the entire line is treated as a comment.
- One or more blank lines

▼ To Create a Profile

1. Use a text editor to create a text file. Name the file descriptively. Or, open a sample profile in the JumpStart directory that you created.

Note – Ensure that the name of the profile reflects how you intend to use the profile to install the Solaris software on a system. For example, you might name the profiles `basic_install`, `eng_profile`, or `user_profile`.

2. Add profile keywords and values to the profile.

For a list of profile keywords and values, see “Profile Keywords and Values” on page 194.

Note – Profile keywords and their values are case sensitive.

3. Save the profile in the JumpStart directory.
4. Ensure that `root` owns the profile and that the permissions are set to 644.
5. Test the profile (optional).
“Testing a Profile” on page 153 contains information about testing profiles.

Profile Examples

The following examples of profiles show how to use different profile keywords and profile values to control how the Solaris software is installed on a system. “Profile Keywords and Values” on page 194 contains a description of profile keywords and values.

EXAMPLE 23-2 Mounting Remote File Systems and Adding and Deleting Packages

```
# profile keywords      profile values
# -----
install_type          initial_install1
system_type           standalone2
partitioning          default3
filesys                any 512 swap # specify size of /swap
cluster               SUNWCprog4
package               SUNWman delete5
cluster               SUNWCacc
```

1. The `install_type` keyword is required in every profile.
2. The `system_type` keyword defines that the system is to be installed as a standalone system.
3. The file system slices are determined by the software to be installed with the value `default`. The size of `swap` is set to 512 Mbytes and is installed on any disk, value `any`.
4. The Developer Solaris Software Group, `SUNWCprog`, is installed on the system.
5. If the standard man pages are mounted from the file server, `s_ref`, on the network, the man page packages are not to be installed on the system. The packages that contain the System Accounting utilities are selected to be installed on the system.

EXAMPLE 23-3 Specifying Where to Install File Systems

```
# profile keywords      profile values
# -----
install_type          initial_install
system_type           standalone
partitioning          explicit1
filesys                c0t0d0s0 auto /
filesys                c0t3d0s1 auto swap
filesys                any auto usr
cluster               SUNWCall2
```

1. The file system slices are determined by the `filesys` keywords, value `explicit`. The size of root (/) is based on the selected software, value `auto`, and is installed on `c0t0d0s0`. The size of `swap` is set to the necessary size and is installed on `c0t3d0s1`. `usr` is based on the selected software and the installation program determines where `usr` is installed, based on the value `any`.
2. The Entire Solaris Software Group, `SUNWCall`, is installed on the system.

EXAMPLE 23-4 Reallocating Disk Space for an Upgrade

```
# profile keywords      profile values
# -----
install_type          upgrade1
root_device           c0t3d0s22
backup_media          remote_filesystem timber:/export/scratch3
```


EXAMPLE 23-4 Reallocating Disk Space for an Upgrade (Continued)

```
layout_constraint      c0t3d0s2 changeable 1004
layout_constraint      c0t3d0s4 changeable
layout_constraint      c0t3d0s5 movable
package                SUNWbcp delete5
package                SUNWxwman add6
cluster                SUNWCacc add
locale                 de7
```

1. The profile upgrades a system by reallocating disk space. In this example, disk space must be reallocated because some file systems on the system did not have enough space for the upgrade.
2. The root file system on `c0t3d0s2` is upgraded.
3. A remote system that is named `timber` is to be used to back up data during the disk space reallocation.
4. The `layout_constraint` keywords designate that auto-layout can perform the following when auto-layout attempts to reallocate disk space for the upgrade.
 - Change slices 2 and 4. The slices can be moved to another location and the size can be changed.
 - Move slice 5. The slice can be moved to another location but its size cannot change.
5. The binary compatibility package, `SUNWbcp`, is not installed on the system after the upgrade.
6. The code ensures that the X Window System man pages and the System Accounting Utilities are to be installed if they are not already installed on the system. All packages already on the system are automatically upgraded.
7. The German localization packages are to be installed on the system.

Testing a Profile

After you create a profile, use the `pinstall(1M)` command to test the profile. Test the profile before you use the profile to install or upgrade a system. Testing a profile is especially useful when you are creating upgrade profiles that reallocate disk space.

By looking at the installation output that is generated by `pinstall`, you can quickly determine if a profile works as you intended. For example, use the profile to determine if a system has enough disk space to upgrade to a new release of the Solaris software before you perform the upgrade on that system.

`pinstall` enables you to test a profile against the following:

- The system's disk configuration where `pinstall` is being run.

- Other disk configurations. You use a disk configuration file that represents a structure of a disk, for example, a disk's bytes/sector, flags, and slices. Creating disk configuration files is described in "Creating Disk Configuration Files" on page 167.

Note – You cannot use a disk configuration file to test a profile you intend to use to upgrade a system. Instead, you must test the profile against the system's actual disk configuration and the software that is currently installed on that system.

▼ To Create a Temporary Solaris 9 Environment to Test a Profile

To test a profile for a particular Solaris release successfully and accurately, you must test a profile within the Solaris environment of the same release. For example, if you want to test a Solaris 9 initial installation profile, run the `pfinstall` command on a system that is running Solaris 9.

You need to create a temporary installation environment if you are testing a profile under one of the following conditions:

- You want to test a Solaris 9 upgrade profile on a system that is running a previous version of the Solaris software.
- You do not have a Solaris 9 system installed yet to test Solaris 9 initial installation profiles.

1. Boot a system from an image of one of the following:

- Solaris 9 DVD
- Solaris 9 Software 1 of 2 CD

Note – If you want to test an upgrade profile, boot the system that you are upgrading.

2. Respond to the system identification questions.

3. Exit from the installation program.

- If you are using the Solaris 9 DVD, type `!` at the following prompt:

```
Solaris Web Start will assist you in installing software for Solaris.  
<Press ENTER to continue> {"!" exits}
```
- If you are using the Solaris 9 Software 1 of 2 CD, select the Solaris `suninstall` program as the program to install the Solaris 9 operating environment. On the first screen that is displayed, select `Exit`.

4. Execute the `pfinstall` command from the shell. For details about using the `pfinstall` command, see Step 7 in “To Test a Profile” on page 155.

▼ To Test a Profile

1. Locate a system on which to test the profile.

If you are testing an upgrade profile, you must test the profile on the actual system that you intend to upgrade.

2. Use the following decision table to determine what to do next.

Test Scenario	Instructions
Test an initial installation profile and have a system that is running the Solaris 9 software	Become superuser on the system and go to Step 5.
Test an upgrade profile, or you do not have a system that is running Solaris 9 to test an initial installation profile	Create a temporary Solaris 9 environment to test the profile. For details, see “To Create a Temporary Solaris 9 Environment to Test a Profile” on page 154. Then, go to Step 3.

3. Create a temporary mount point.

```
# mkdir /tmp/mnt
```

4. Mount the directory that contains the profile or profiles that you want to test.

Mount Scenario	Typing Instructions
Mount a remote NFS file system for systems on the network	<code>mount -F nfs server_name:path /tmp/mnt</code>
Mount a UFS-formatted diskette	<code>mount -F ufs /dev/diskette /tmp/mnt</code>

5. To test the profile with a specific system memory size, set `SYS_MEMSIZE` to the specific memory size in Mbytes.

```
# SYS_MEMSIZE=memory_size
# export SYS_MEMSIZE
```

6. Did you mount a directory in Step 4?

- If yes, change the directory to `/tmp/mnt`.

```
# cd /tmp/mnt
```

- If no, change the directory to where the profile is located, which is usually the JumpStart directory.

```
# cd jumpstart_dir_path
```

7. Test the profile with the `pinstall(1M)` command.

```
# /usr/sbin/install.d/pinstall -D:-d disk_config_file [-c path] profile
```



Caution – You *must* include the `-d` or `-D` option. If you do not include one of these options, `pinstall` uses the profile you specify to install the Solaris 9 software. All of the data on the system is overwritten.

<code>-D</code>	<code>pinstall</code> uses the current system's disk configuration to test the profile. You must use the <code>-D</code> option to test an upgrade profile.
<code>-d disk_config_file</code>	<code>pinstall</code> uses the disk configuration file, <code>disk_config_file</code> , to test the profile. If <code>disk_config_file</code> is not located in the directory where <code>pinstall</code> is run, you must specify the path. For instructions on how to create a disk configuration file, see "Creating Disk Configuration Files" on page 167. Note – You cannot use the <code>-d disk_config_file</code> option with an upgrade profile, <code>install_type upgrade</code> . You must always test an upgrade profile against a system's disk configuration, that is, you must use the <code>-D</code> option.
<code>-c path</code>	The path to the Solaris 9 software image. You use this option, for example, if the system is using Volume Manager to mount the Solaris 9 Software 1 of 2 CD for your platform. Note – The <code>-c</code> option is not required if you booted from a Solaris 9 DVD or a Solaris 9 Software 1 of 2 CD image for your platform. The DVD or CD image is mounted on <code>/cdrom</code> as part of the booting process.
<code>profile</code>	The name of the profile to test. If <code>profile</code> is not in the directory where <code>pinstall</code> is being run, you must specify the path.

Profile Test Examples

The following example shows how to use `pinstall` to test a profile that is named `basic_prof`. The profile is tested against the disk configuration on a system on which the Solaris 9 software is installed. The `basic_prof` profile is located in the `/jumpstart` directory, and the path to the Solaris 9 Software 1 of 2 CD image is specified because Volume Manager is being used.

EXAMPLE 23–5 Profile Test Using a Solaris 9 System

```
# cd /jumpstart
# /usr/sbin/install.d/pinstall -D -c /cdrom/pathname basic_prof
```

The following example shows how to use `pfinstall` to test the profile that is named `basic_prof` on a Solaris 9 system. The test is performed against the `535_test` disk configuration file. The test checks for 64 Mbytes of system memory. This example uses a Solaris 9 Software 1 of 2 CD image that is located in the `/export/install` directory.

EXAMPLE 23-6 Profile Test Using a Disk Configuration File

```
# SYS_MEMSIZE=64
# export SYS_MEMSIZE
# /usr/sbin/install.d/pfinstall -d 535_test -c /export/install basic_prof
```

Validating the rules File

Before you can use a profile and rules file, you must run the check script to validate that the files are set up correctly. If all rules and profiles are correctly set up, the `rules.ok` file is created, which is required by the custom JumpStart installation software to match a system to a profile.

Table 23-3 describes what the check script does.

TABLE 23-3 What Happens When You Use the check Script

Stage	Description
1	The <code>rules</code> file is checked for syntax. check verifies that the rule keywords are legitimate and that the <i>begin</i> , <i>class</i> , and <i>finish</i> fields are specified for each rule. The <i>begin</i> and <i>finish</i> fields can consist of a minus sign (-) instead of a file name.
2	If no errors are found in the <code>rules</code> file, each profile that is specified in the rules is checked for syntax.
3	If no errors are found, check creates the <code>rules.ok</code> file from the <code>rules</code> file, removes all comments and blank lines, retains all rules, and adds the following comment line at the end: # version=2 checksum=num

▼ To Validate the rules File

1. Ensure that the check script is located in the JumpStart directory.

Note – The check script is in the `Solaris_9/Misc/jumpstart_sample` directory on the Solaris 9 DVD or on the Solaris 9 Software 1 of 2 CD.

2. Change the directory to the JumpStart directory.

3. Run the check script to validate the rules file:

```
$ ./check [-p path -r file_name]
```

<code>-p path</code>	Validates the <code>rules</code> by using the check script from the Solaris 9 software image instead of the check script from the system you are using. <code>path</code> is the image on a local disk or a mounted Solaris 9 DVD or a Solaris 9 Software 1 of 2 CD. Use this option to run the most recent version of <code>check</code> if your system is running a previous version of Solaris.
<code>-r file_name</code>	Specifies a rules file other than the one that is named <code>rules</code> . Using this option, you can test the validity of a rule before you integrate the rule into the <code>rules</code> file.

As the check script runs, the script reports the checking of the validity of the `rules` file and each profile. If no errors are encountered, the script reports: The custom JumpStart configuration is ok.

4. Ensure that root owns the rules.ok file and that the permissions are set to 644.

After you validate the `rules` file, you can learn more about optional custom JumpStart features in Chapter 24. You can learn about performing custom JumpStart installations in Chapter 26.

Using Optional Custom JumpStart Features (Tasks)

This chapter describes the optional features that are available to create additional custom JumpStart installation tools.

- “Creating Begin Scripts” on page 159
- “Creating Finish Scripts” on page 161
- “Creating a Compressed Configuration File” on page 166
- “Creating Disk Configuration Files” on page 167
- “Using a Site-Specific Installation Program” on page 169

Creating Begin Scripts

A begin script is a user-defined Bourne shell script that you specify in the `rules` file. A begin script performs tasks before the Solaris software is installed on a system. You can use begin scripts only when using custom JumpStart to install the Solaris software.

Use a begin script to perform one of the following tasks:

- Create derived profiles
- Back up files before upgrading

Important Information About Begin Scripts

- Do not specify something in the script that would prevent the mounting of file systems onto `/a` during an initial or upgrade installation. If the JumpStart program cannot mount the file systems onto `/a`, an error occurs and installation fails.
- Output from the begin script is deposited in `/var/sadm/begin.log`.
- Ensure that `root` owns the begin script and that the permissions are set to 644.

- You can use custom JumpStart Environment variables in your begin scripts. For a list of environment variables, see “Custom JumpStart Environment Variables” on page 217.
- Save begin scripts in the JumpStart directory.

Creating Derived Profiles With a Begin Script

A derived profile is a profile that is dynamically created by a begin script during a custom JumpStart installation. Derived profiles are needed when you cannot set up the `rules` file to match specific systems to a profile. For example, you might need to use derived profiles for identical system models that have different hardware components, such as systems that contain different frame buffers.

To set up a rule to use a derived profile, you must perform the following tasks:

- Set the profile field to an equal sign (=) instead of a profile.
- Set the begin field to a begin script that creates a derived profile that depends on the system on which you intend to install Solaris.

When a system matches a rule with the profile field equal to an equal sign (=), the begin script creates the derived profile that is used to install the Solaris software on the system.

The following is an example of a begin script that creates the same derived profile every time. You can write a begin script to create different derived profiles that depend on the evaluation of rules.

EXAMPLE 24–1 Begin Script That Creates a Derived Profile

```
#!/bin/sh
echo "install_type      initial_install"    > ${SI_PROFILE}
echo "system_type      standalone"      >> ${SI_PROFILE}
echo "partitioning     default"         >> ${SI_PROFILE}
echo "cluster          SUNWCprog"       >> ${SI_PROFILE}
echo "package          SUNWman    delete" >> ${SI_PROFILE}
echo "package          SUNWolman  delete" >> ${SI_PROFILE}
echo "package          SUNWxwman  delete" >> ${SI_PROFILE}
```

In the example, the begin script must use the `SI_PROFILE` environment variable for the name of the derived profile, which is set to `/tmp/install.input` by default.

Note – If a begin script is used to create a derived profile, ensure the script does not have any errors. A derived profile is not verified by the check script because derived profiles are not created until the execution of the begin script.

Creating Finish Scripts

A finish script is a user-defined Bourne shell script that you specify in the `rules` file. A finish script performs tasks after the Solaris software is installed on a system, but before the system reboots. You can use finish scripts only when using custom JumpStart to install Solaris.

Tasks that you can perform with a finish script include the following:

- Add files
- Add individual packages or patches in addition to the ones that are installed in a particular software group
- Customize the root environment
- Set the system's root password
- Install additional software

Important Information About Finish Scripts

- The Solaris `suninstall` program mounts the system's file systems on `/a`. The file systems remain mounted on `/a` until the system reboots. You can use the finish script to add, change, or remove files from the newly installed file system hierarchy by modifying the file systems that are respective to `/a`.
- Output from the finish script is deposited in `/var/sadm/finish.log`.
- Ensure that `root` owns the finish script and that the permissions are set to `644`.
- You can use custom JumpStart Environment variables in your finish scripts. For a list of environment variables, see "Custom JumpStart Environment Variables" on page 217.
- Save finish scripts in the JumpStart directory.

▼ To Add Files With a Finish Script

Through a finish script, you can add files from the JumpStart directory to an already installed system. You can add the files because the JumpStart directory is mounted on the directory that is specified by the `SI_CONFIG_DIR` variable. The directory is set to `/tmp/install_config` by default.

Note – You can also replace files by copying files from the JumpStart directory to already existing files on the installed system.

1. Copy all of the files that you are adding to the installed system to the JumpStart directory.
2. Insert the following line in the finish script for each file that you want to be copied to the newly installed file system hierarchy:

```
cp ${SI_CONFIG_DIR}/file_name /a/path_name
```

For example, assume you have a special application, `site_prog`, developed for all users at your site. If you place a copy of `site_prog` into the JumpStart directory, the following line in a finish script copies `site_prog` from the JumpStart directory into a system's `/usr/bin` directory:

```
cp ${SI_CONFIG_DIR}/site_prog /a/usr/bin
```

Adding Packages or Patches With a Finish Script

You can create a finish script to automatically add packages or patches after the Solaris software is installed on a system. By adding packages with a finish script, you reduce time and ensure consistency in what packages and patches are installed on different systems at your site.

When you use the `pkgadd(1M)` or `patchadd(1M)` commands in finish scripts, use the `-R` option to specify `/a` as the root path.

Example 24–2 shows an example of a finish script that adds packages.

EXAMPLE 24–2 Adding Packages With a Finish Script

```
#!/bin/sh

BASE=/a
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin

mkdir ${MNT}
mount -f nfs sherlock:/export/package ${MNT}1
cat >${ADMIN_FILE} <<DONT_ASK2
mail=root
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=ask
setuid=nocheck
```

EXAMPLE 24–2 Adding Packages With a Finish Script (Continued)

```
conflict=nocheck
action=nocheck
basedir=default
DONT_ASK

/usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNWxyz3
umount ${MNT}
rmdir ${MNT}
```

1. Mounts a directory on a server that contains the package to install.
2. Creates a temporary package administration file, `admin`, to force the `pkgadd(1M)` command not to perform checks or prompt for questions when installing a package. Use the temporary package administration file to maintain a hands-off installation when you are adding packages.
3. Adds the package by using the `-a` option, specifying the package administration file, and the `-R` option, specifying the root path.

Note – In the past, the `chroot(1M)` command was used with the `pkgadd` and `patchadd` commands in the finish script environment. In rare instances, some packages or patches do not work with the `-R` option. You must create a dummy `/etc/mnttab` file in the `/a` root path before issuing the `chroot` command.

To create a dummy `/etc/mnttab` file, add the following line to your finish script:

```
cp /etc/mnttab /a/etc/mnttab
```

Customizing the Root Environment With a Finish Script

You can also use finish scripts to customize files that are already installed on a system. For example, the finish script in Example 24–3 customizes the root environment by appending information to the `.cshrc` file in the root (`/`) directory.

EXAMPLE 24–3 Customizing the Root Environment With a Finish Script

```
#!/bin/sh
#
# Customize root's environment
#
echo "***adding customizations in /.cshrc"
test -f a/.cshrc || {
cat >> a/.cshrc <<EOF
set history=100 savehist=200 filec ignoreeof prompt="\$user@`uname -n`> "
alias cp cp -i
```

EXAMPLE 24-3 Customizing the Root Environment With a Finish Script (Continued)

```
alias mv mv -i
alias rm rm -i
alias ls ls -FC
alias h history
alias c clear
unset autologout
EOF
}
```

Setting a System's Root Password With a Finish Script

After the Solaris software is installed on a system, the system reboots. Before the boot process is completed, the system prompts for the root password. Until someone types a password, the system cannot finish booting.

A finish script that is named `set_root_pw` is saved in the `auto_install_sample` directory. The finish script shows how to set the root password automatically, without prompting. `set_root_pw` is shown in Example 24-4.

EXAMPLE 24-4 Setting the System's Root Password With a Finish Script

```
#!/bin/sh
#
#      @(#)set_root_pw 1.4 93/12/23 SMI
#
# This is an example Bourne shell script to be run after installation.
# It sets the system's root password to the entry defined in PASSWD.
# The encrypted password is obtained from an existing root password entry
# in /etc/shadow from an installed machine.

echo "setting password for root"

# set the root password
PASSWD=dKO5IBkSF42lw
#create a temporary input file1
cp /a/etc/shadow /a/etc/shadow.orig2

mv /a/etc/shadow /a/etc/shadow.orig
nawk -F: '{
    if ( $1 == "root" )3
        printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n", $1,passwd,$3,$4,$5,$6,$7,$8,$9
    else
        printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n", $1,$2,$3,$4,$5,$6,$7,$8,$9
}' passwd="$PASSWD" /a/etc/shadow.orig > /a/etc/shadow
#remove the temporary file
rm -f /a/etc/shadow.orig4
# set the flag so sysidroot won't prompt for the root password
sed -e 's/0 # root/1 # root/' ${SI_SYS_STATE} > /tmp/state.$$5
```

EXAMPLE 24-4 Setting the System's Root Password With a Finish Script (Continued)

```
mv /tmp/state.$$ ${SI_SYS_STATE}
```

1. Sets the variable `PASSWD` to an encrypted root password that is obtained from an existing entry in a system's `/etc/shadow` file.
2. Creates a temporary input file of `/a/etc/shadow`.
3. Changes the root entry in the `/etc/shadow` file for the newly installed system by using `$PASSWD` as the password field.
4. Removes the temporary `/a/etc/shadow` file.
5. Changes the entry from 0 to a 1 in the state file so that the user is not prompted for the root password. The state file is accessed by using the variable `SI_SYS_STATE`, which has a value currently of `/a/etc/.sysIDtool.state`. To avoid problems with your scripts if this value changes, always reference this file by using `$SI_SYS_STATE`. The `sed` command that is shown here contains a tab character after the 0 and after the 1.

Note – If you set the system's root password with a finish script, users might attempt to discover the root password from the encrypted password in your finish script. Ensure that you safeguard against users who might try to determine the root password.

Installing Software With Web Start Installation Programs With Finish Scripts

You can use finish scripts to install additional software after the Solaris operating environment is installed. Some software programs are installed by the Solaris Web Start program, which prompts you to enter information during the installation. To maintain a hands-off installation, you can run the Solaris Web Start program with the `-nodisplay` or `-noconsole` options.

TABLE 24-1 Solaris Web Start Options

Option	Description
<code>-nodisplay</code>	Runs the installer without a graphic user interface. Use the default product installation unless the installation was modified by the <code>-locales</code> option.

TABLE 24-1 Solaris Web Start Options (Continued)

Option	Description
-noconsole	Runs the installation without any interactive text console device. Useful when paired with -nodisplay for UNIX script use.

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

Creating a Compressed Configuration File

Rather than using the `add_install_client` command to specify the location of the custom JumpStart configuration files, you can specify the location of the files when you boot the system. However, you can only specify the name of one file when you issue the `boot` command. As a result, you must compress all of the custom JumpStart configuration files into one file. The compressed configuration file can be one of the following types:

- `tar`
- Compressed `tar`
- `zip`
- `bzip tar`

▼ To Create a Compressed Configuration File

1. Change the directory to the JumpStart directory on the profile server.

```
# cd jumpstart_dir_path
```

2. Use a compression tool to compress the custom JumpStart configuration files into one file.

Note – The compressed configuration file cannot contain relative paths. The custom JumpStart configuration files must be in the same directory as the compressed file.

The compressed configuration file must contain the following files:

- `Profile`
- `rules`

- `rules.ok`

You can also include the `sysidcfg` file in the compressed configuration file.

3. **Save the compressed configuration file on an NFS server, an HTTP server, or on a local hard disk.**

Compressed Configuration File Example

The following example shows how to use the `tar` command to create a compressed configuration file that is named `config.tar`. The custom JumpStart configuration files are located in the `/jumpstart` directory.

EXAMPLE 24-5 Creating a Compressed Configuration File

```
# cd /jumpstart
# tar -cvf config.tar *
a profile 1K
a rules 1K
a rules.ok 1K
a sysidcfg 1K
```

Creating Disk Configuration Files

This section describes how to create single-disk and multiple-disk configuration files. Disk configuration files enable you to use `pinstall(1M)` from a single system to test profiles against different disk configurations.

▼ To Create a Disk Configuration File

1. **Locate a system with a disk you want to test.**
2. **Become superuser.**
3. **Create a single-disk configuration file by redirecting the output of the `prtvtoc(1M)` command to a file.**

```
# prtvtoc /dev/rdisk/device_name >disk_config_file
```

<i>/dev/rdisk/device_name</i>	The device name of the system's disk. <i>device_name</i> must be in the form <i>cwt.xdys2</i> or <i>cxdys2</i> .
<i>disk_config_file</i>	The name of the disk configuration file.

4. Determine if you are testing the installation of Solaris software on multiple disks.

- If no, stop. You are finished.
- If yes, concatenate the single-disk configuration files and save the output in a new file.

```
# cat disk_file1 disk_file2 >multi_disk_config
```

The new file becomes the multiple-disk configuration file, as in the following example:

```
# cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
```

5. Determine if the target numbers in the disk device names are unique in the multiple-disk configuration file that you created in the previous step.

- If yes, stop. You are finished.
- If no, open the file with a text editor and make the target numbers unique in the disk device names.

For example, assume that the file contains the same target number, *t0*, for different disk device names, as shown here:

```
* /dev/rdsk/c0t0d0s2 partition map
...
* /dev/rdsk/c0t0d0s2 partition map
```

Change the second target number to *t2*, as shown here:

```
* /dev/rdsk/c0t0d0s2 partition map
...
* /dev/rdsk/c0t2d0s2 partition map
```

Disk Configuration File Example

The following example shows how to create a single-disk configuration file, *104_test*, on a system with a 104-Mbyte disk.

EXAMPLE 24-6 Creating a Disk Configuration File

You redirect the output of the `prtvtoc` command to a single-disk configuration file that is named *104_test*:

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

The contents of the *104_test* file resemble the following:

EXAMPLE 24-6 Creating a Disk Configuration File (Continued)

```
* /dev/rdisk/c0t3d0s2 partition map
*
* Dimensions:
*   512 bytes/sector
*   72 sectors/track
*   14 tracks/cylinder
*   1008 sectors/cylinder
*   2038 cylinders*   2036 accessible cylinders
* Flags:
*   1: unmountable
*   10: read-only
*
*
* Partition  Tag  Flags      First   Sector   Last   Mount Directory
*   1         2    00         0     164304   164303 /
*   2         5    00         0    2052288 2052287
*   3         0    00    164304   823536   987839 /disk2/b298
*   5         0    00    987840   614880  1602719 /install/298/sparc/work
*   7         0    00   1602720  449568  2052287 /space
```

You have created disk configuration files. “Testing a Profile” on page 153 contains information about using disk configuration files to test profiles.

Using a Site-Specific Installation Program

You can also use begin and finish scripts to create your own installation program to install Solaris software.

When you specify a minus sign (-) in the profile field, begin and finish scripts control how Solaris software is installed on a system instead of the profile and the Solaris suninstall program.

For example, if the following rule matches a system, the `x_install.beg` begin script and the `x_install.fin` finish script install Solaris software on the system that is named `clover`:

```
hostname clover x_install.beg - x_install.fin
```


Creating Custom Rule and Probe Keywords (Tasks)

This chapter provides information and procedures for creating your own custom rule and probe keywords.

- “Probe Keywords” on page 171
- “Creating a custom_probes File” on page 172
- “Validating the custom_probes File” on page 175

Probe Keywords

To understand what a probe keyword is, you first need to recall what a rule keyword is. A rule keyword is a predefined lexical unit or word that describes a general system attribute, such as host name, `hostname`, or memory size, `memsize`. Rule keywords and the values that are associated with them enable you to match a system that has the same attribute to a profile. This match of a system’s attributes defines how the Solaris software is to be installed on each system in the group.

Custom JumpStart environment variables, which you use in begin and finish scripts, are set on demand. For example, information about which operating system is already installed on a system is only available in `SI_INSTALLED` after the `installed` rule keyword is used.

In some situations, you might need to extract the same information in a begin or finish script for a purpose other than to match a system and run a profile. Probe keywords provide the solution. Probe keywords extract attribute information and remove the need for you to set up a matching condition and run a profile.

For a list of probe keywords and values, see “Probe Keywords and Values” on page 220.

Creating a custom_probes File

If the rule and probe keywords that are described in “Rule Keywords and Values” on page 189 and “Probe Keywords and Values” on page 220 are not precise enough for your needs, you can define your own custom rule or probe keywords by creating a custom_probes file.

The custom_probes file is a Bourne shell script that contains two types of functions. You must save the custom_probes file in the same JumpStart directory where you saved the rules file. The two types of functions that you can define in a custom_probes file are as follows:

- **Probe** – Gathers the information you want or does the actual work and sets a corresponding SI_ environment variable that you define. Probe functions become probe keywords.
- **Comparison** – Calls a corresponding probe function, compares the output of the probe function, and returns 0 if the keyword matches or 1 if the keyword does not match. Comparison functions become rule keywords.

Syntax of the custom_probes File

The custom_probes file can contain any valid Bourne shell command, variable, or algorithm.

Note – You can define probe and comparison functions that require a single argument in the custom_probes file. When you use the corresponding custom probe keyword in the rules file, the argument after the keyword is interpreted (as \$1).

When you use the corresponding custom rule keyword in the rules file, the argument is interpreted as starting after the keyword and ending before the next && or begin script, whichever comes first.

The custom_probes file must meet the following requirements:

- Have the name custom_probes
- Have root as its owner
- Be executable and have permissions set to 755
- Contain at least one probe function and one corresponding comparison function

To improve clarity and organization, define all probe functions first, at the top of the file, followed by all comparison functions.

Syntax of Function Names in `custom_probes`

The name of a probe function must begin with `probe_`. The name of a comparison function must begin with `cmp_`.

Functions that begin with `probe_` define new probe keywords. For example, the function `probe_tcx` defines the new probe keyword `tcx`. Functions that begin with `cmp_` define new rule keywords. For example, `cmp_tcx` defines the new rule keyword `tcx`.

▼ To Create a `custom_probes` File

1. Use a text editor to create a Bourne shell script text file. Name the file `custom_probes`.
2. In the `custom_probes` text file, define your probe and comparison functions.

Note – You can define probe and comparison functions that require arguments in the `custom_probes` file. When you use the corresponding custom probe keyword in the rules file, the arguments after the keyword are interpreted in sequence (as `$1`, `$2`, and so on).

When you use the corresponding custom rule keyword in the `rules` file, the arguments are interpreted in sequence after the keyword and before the next `&&` or `begin script`, whichever comes first.

3. Save the `custom_probes` file in the `JumpStart` directory next to the `rules` file.
4. Ensure that `root` owns the `rules` file and that the permissions are set to `644`.

Examples of a `custom_probes` File and Keyword

You can find additional examples of probe and comparison functions in the following directories:

- `/usr/sbin/install.d/chkprobe` on a system that has the Solaris software installed
- `/Solaris_9/Tools/Boot/usr/sbin/install.d/chkprobe` on the Solaris 9 DVD or on the Solaris 9 Software 1 of 2 CD

The following `custom_probes` file contains a probe and comparison function that tests for the presence of a TCX graphics card.

EXAMPLE 25-1 custom_probes File

```
#!/bin/sh
#
# custom_probe script to test for the presence of a TCX graphics card.
#
#
# PROBE FUNCTIONS
#
probe_tcx() {
    SI_TCX=`modinfo | grep tcx | nawk '{print $6}'`
    export SI_TCX
}

#
# COMPARISON FUNCTIONS
#
cmp_tcx() {
    probe_tcx

    if [ "X${SI_TCX}" = "X${1}" ]; then
        return 0
    else
        return 1
    fi
}
```

The following example `rules` file shows the use of the probe keyword that is defined in the preceding example, `tcx`. If a TCX graphics card is installed and found in a system, `profile_tcx` is run. Otherwise, `profile` is run.

Note – Always place probe keywords at or near the beginning of the `rules` file to ensure that the keywords are read and run before other rule keywords that might rely on the probe keywords.

EXAMPLE 25-2 Custom Probe Keyword Used in a `rules` File

```
probe tcx
tcx    tcx    -    profile_tcx    -
any    any    -    profile        -
```

Validating the `custom_probes` File

Before you can use a profile, rules, and `custom_probes` file, you must run the check script to validate that the files are set up correctly. If all profiles, rules, and probe and comparison functions are correctly set up, the `rules.ok` and `custom_probes.ok` files are created. Table 25–1 describes what the check script does.

TABLE 25–1 What Happens When You Use the check Script

Stage	Description
1	check searches for a <code>custom_probes</code> file.
2	If the file exists, check creates the <code>custom_probes.ok</code> file from the <code>custom_probes</code> file, removes all comments and blank lines, and retains all Bourne shell commands, variables, and algorithms. Then, check adds the following comment line at the end: <code># version=2 checksum=num</code>

▼ To Validate the `custom_probes` File

1. Verify that the check script is located in the JumpStart directory.

Note – The check script is in the `Solaris_9/Misc/jumpstart_sample` directory on the Solaris 9 DVD or on the Solaris 9 Software 1 of 2 CD.

2. Change to the JumpStart directory.
3. Run the check script to validate the rules and `custom_probes` files.

```
$ ./check [-p path -r file_name]
```

`-p path`

Validates the `custom_probes` file by using the check script from the Solaris 9 software image for your platform instead of the check script from the system you are using. `path` is the image on a local disk or a mounted Solaris 9 DVD or Solaris 9 Software 1 of 2 CD.

Use this option to run the most recent version of check if your system is running a previous version of Solaris.

`-r file_name` Specifies a file name other than the one that is named `custom_probes`. By using the `-r` option, you can test the validity of a set of functions before integrating the functions into the `custom_probes` file.

As the check script runs, the script reports the validity of the `rules` and `custom_probes` files and each profile. If no errors are encountered, the script reports: "The custom JumpStart configuration is ok" and creates the `rules.ok` and `custom_probes.ok` files in the JumpStart directory.

4. Determine if the `custom_probes.ok` file is executable.

- If yes, go to Step 5.
- If no, type the following command:

```
chmod +x custom_probes
```

5. Ensure that `root` owns the `custom_probes.ok` file and that the permissions are set to 755.

Performing a Custom JumpStart Installation (Tasks)

This chapter describes how to perform a custom JumpStart installation. You need to follow these procedures on the system on which you intend to install the Solaris 9 software.

Performing a Custom JumpStart Installation

Task Map: Setting Up a System for a Custom JumpStart Installation

During a custom JumpStart installation, the JumpStart program attempts to match the system that is being installed to the rules in the `rules.ok` file. The JumpStart program reads the rules from the first rule through the last. A match occurs when the system that is being installed matches all the system attributes that are defined in the rule. As soon as a system matches a rule, the JumpStart program stops reading the `rules.ok` file and begins to install the system, based on the matched rule's profile.

TABLE 26-1 Task Map: Setting Up a System for a Custom JumpStart Installation

Task	Description	For Instructions
Check if the system is supported	Check the hardware documentation for system support in the Solaris 9 environment.	<i>Solaris 9 Sun Hardware Platform Guide</i>
Check if the system has enough disk space for the Solaris 9 software	Verify that you have planned enough space to install the Solaris software on your system.	Chapter 5
(Optional) Preconfigure system configuration information	You can use the <code>sysidcfg</code> file or the name service to preconfigure installation information for a system. If you preconfigure system information, the installation program does not prompt you to supply the information during the installation.	Chapter 7
Prepare the system for custom JumpStart installation	Create and validate a <code>rules</code> file and profile files.	Chapter 23
(Optional) Prepare optional custom JumpStart features	If you are using begin scripts, finish scripts, or other optional features, prepare the scripts or files.	Chapter 24 and Chapter 25
(Optional) Set up the system to install over the network	To install a system from a remote Solaris 9 DVD or Solaris 9 Software CD image, you need to set up the system to boot and install from an install server or a boot server.	Chapter 12
Install or upgrade	Boot the system to initiate the installation or upgrade.	"To Perform an Installation or Upgrade With the Custom JumpStart Program" on page 178

▼ To Perform an Installation or Upgrade With the Custom JumpStart Program

1. If the system is part of a network, ensure that an Ethernet connector or similar network adapter is attached to your system.

2. If you are installing a system that is connected through a `tip(1)` line, ensure that your window display is at least 80 columns wide and 24 rows long.
To determine the current dimensions of your `tip` window, use the `stty(1)` command.
3. If you are using the system's DVD-ROM or CD-ROM drive to install the Solaris 9 software, insert the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD in the drive.
4. If you are using a profile diskette, insert the profile diskette in the system's diskette drive.
5. Boot the system.

- If the system is new, out-of-the-box, turn on the system.
- If you want to install or upgrade an existing system, shut down the system. At the `ok` prompt, type the following command:

```
ok boot cdrom:net - install [url:ask] [dhcp] [nowin]
```

<code>cdrom</code>	<p>Specifies to boot from a CD or a DVD.</p> <p>For a system with an older EEPROM, replace <code>cdrom</code> with <code>sd(0,6,2)</code> to boot from the system's CD-ROM or DVD-ROM drive.</p>
<code>net</code>	<p>Specifies to boot from an install server on the network.</p>
<code>url</code>	<p>Specifies the location of the custom JumpStart files. You can specify a URL for files that are located in the following places:</p> <ul style="list-style-type: none"> ■ Local hard disk <ul style="list-style-type: none"> <code>file://jumpstart_dir_path/compressed_config_file</code> ■ NFS server <ul style="list-style-type: none"> <code>nfs://server_name:IP_address/jumpstart_dir_path/compressed_config_file</code> ■ HTTP server <ul style="list-style-type: none"> <code>http://server_name:IP_address/jumpstart_dir_path/compressed_config_file&proxy_info</code> <p>If you placed a <code>sysidcfg</code> file in the compressed configuration file, you must specify the IP address of the server that contains the file, as in the following example:</p> <pre>http://131.141.2.32/jumpstart/config.tar</pre> <p>If you saved the compressed configuration file on an HTTP server that is behind a firewall, you must use a proxy specifier during boot. You do not need to specify an IP address for the server that contains the file. You must specify an IP address for the proxy server, as in the following example:</p> <pre>http://www.shadow.com/jumpstart/config.tar&proxy=131.141.6.151</pre>

ask	<p>Specifies that the installation program prompt you to type the location of the compressed configuration file after the system boots and connects to the network.</p> <p>If you bypass the prompt by pressing Return, the installation program interactively configures the network parameters. The installation program then prompts you for the location of the compressed configuration file. If you bypass the prompt by pressing Return, the Solaris <code>suninstall</code> program begins.</p>
dhcp	<p>Specifies to use a DHCP server to obtain network installation information that is needed to boot the system.</p> <p>If you do not specify to use a DHCP server, the system uses the <code>/etc/bootparams</code> file or the name service <code>bootparams</code> database.</p>
nowin	<p>Specifies not to begin the X program. You do not need to use the X program to perform a custom JumpStart installation, so you can reduce the installation time by using the <code>nowin</code> option.</p>

Note – The system checks hardware and system components and your system boots. Booting lasts several minutes.

6. If you did not preconfigure system information in the `sysidcfg` file, when prompted, answer the questions about system configuration.

7. Follow the instructions on the screen to install the software.

When the JumpStart program finishes installing the Solaris software, the system reboots automatically.

After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

- `/var/sadm/system/logs`
- `/var/sadm/install/logs`

Installing With Custom JumpStart (Examples)

This chapter provides an example of setting up and installing Solaris software by using a custom JumpStart installation.

- “Sample Site Setup” on page 181
- “Create an Install Server” on page 183
- “Create a Boot Server for Marketing Systems” on page 184
- “Create a JumpStart Directory” on page 184
- “Share the JumpStart Directory” on page 185
- “Create the Engineering Group’s Profile” on page 185
- “Create the Marketing Group’s Profile” on page 185
- “Update the `rules` File” on page 186
- “Validate the `rules` File” on page 187
- “Set Up Engineering Systems to Install From the Network” on page 187
- “Set Up Marketing Systems to Install From the Network” on page 188
- “Boot the Systems and Install Solaris 9 Software” on page 188

Sample Site Setup

Figure 27–1 shows the site setup for this example.

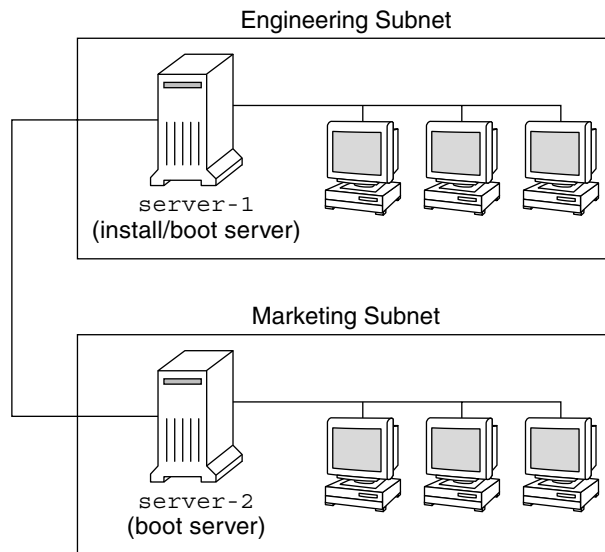


FIGURE 27-1 Sample Site Setup

At this sample site, the conditions are as follows:

- The engineering group is located on its own subnet. This group uses systems for software development.
- The marketing group is located on its own subnet. This group uses systems for running word processors, spreadsheets, and other office productivity tools.
- The site uses NIS. The Ethernet addresses, IP addresses, and host names of the systems are preconfigured in the NIS maps. The subnet mask, date and time, and geographic region for the site are also preconfigured in the NIS maps.

Note – The peripheral devices for the marketing systems are preconfigured in the `sysidcfg` file.

- Both the engineering and marketing systems are to be installed with Solaris 9 software from the network.

Create an Install Server

Because the groups need to install Solaris 9 software from the network, you make `server-1` an install server for both groups. You use the `setup_install_server(1M)` command to copy the images to the `server-1` local disk (in the `/export/install` directory). Copy the images from the Solaris 9 Software CDs and the Solaris 9 Languages CD or from the Solaris 9 DVD.

You must copy the image from the disc to an empty directory, in these examples the `sparc_9` directory.

EXAMPLE 27-1 Copying the Solaris 9 CDs

Insert the Solaris 9 Software 1 of 2 CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# mkdir -p /export/install/sparc_9
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9
```

Insert the Solaris 9 Software 2 of 2 CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

Insert the Solaris 9 Languages CD in the CD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

EXAMPLE 27-2 Copying the Solaris 9 DVD

Insert the Solaris 9 DVD in the DVD-ROM drive that is attached to `server-1` and type the following commands:

```
server-1# mkdir -p /export/install/sparc_9
server-1# cd /DVD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9
```

Create a Boot Server for Marketing Systems

Systems cannot boot from an install server on a different subnet, so you make `server-2` a boot server on the marketing group's subnet. You use the `setup_install_server(1M)` command to copy the boot software from the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD to the `server-2` local disk (in the `/export/boot` directory).

If you insert the Solaris 9 Software 1 of 2 CD in the CD-ROM drive that is attached to `server-2`, type the following command:

```
server-2# cd /CD_mount_point/Solaris_9/Tools
server-2# ./setup_install_server -b /export/boot
```

If you insert the Solaris 9 DVD in the DVD-ROM drive that is attached to `server-2`, type the following command:

```
server-2# cd /DVD_mount_point/Solaris_9/Tools
server-2# ./setup_install_server -b /export/boot
```

In the `setup_install_server` command, `-b` specifies that `setup_install_server` is to copy the boot information to the directory that is named `/export/boot`.

Create a JumpStart Directory

Now that you have the install server and boot server set up, you create a JumpStart directory on `server-1`. You can use any system on the network. This directory holds files that are required for a custom JumpStart installation of Solaris software. You set up this directory by copying the sample directory from the Solaris 9 DVD image or from the Solaris 9 Software 1 of 2 CD image that has been copied to `/export/install`:

```
server-1# mkdir /jumpstart
server-1# cp -r /export/install/sparc_9/Solaris_9/Misc/jumpstart_sample /jumpstart
```

Share the JumpStart Directory

To make the rules file and profiles accessible to systems on the network, you share the `/jumpstart` directory. To enable the sharing of a directory, you add the following line to the `/etc/dfs/dfstab` file:

```
share -F nfs -o ro,anon=0 /jumpstart
```

Then, at the command line, you type the `shareall` command:

```
server-1# shareall
```

Create the Engineering Group's Profile

For the engineering systems, you create a file that is named `eng_prof` in the `/jumpstart` directory. The `eng_prof` file contains the following entries, which define the Solaris 9 software to be installed on systems in the engineering group:

```
install_type  initial_install1
system_type   standalone2
partitioning  default3
cluster       SUNWCprog4
filesystems   any 512 swap5
```

1. Specifies that the installation is to be treated as an initial installation, as opposed to an upgrade.
2. Specifies that the engineering systems are standalone systems.
3. Specifies that the JumpStart software uses default disk partitioning for installing Solaris software on the engineering systems.
4. Specifies that the Developer System Support software group is to be installed.
5. Specifies that each system in the engineering group is to have 512 Mbytes of swap space.

Create the Marketing Group's Profile

For the marketing systems, you create a file that is named `marketing_prof` in the `/jumpstart` directory. The `marketing_prof` file contains the following entries, which define the Solaris 9 software to be installed on systems in the marketing group:

```
install_type  initial_install1
system_type   standalone2
partitioning  default3
cluster       SUNWCuser4
package       SUNWaudio5
```

1. Specifies that the installation is to be treated as an initial installation, as opposed to an upgrade.
2. Specifies that the marketing systems are standalone systems.
3. Specifies that the JumpStart software is to use default disk partitioning for installing Solaris on the marketing systems.
4. Specifies that the End User Solaris Software Group is to be installed.
5. Specifies that the audio demo software package is to be added to each system.

Update the rules File

Now you must add rules to the `rules` file. The Solaris `suninstall` program uses the rules to select the correct installation (profile) for each system during a custom JumpStart installation.

At this site, each department is located on its own *subnet* and has its own network address. The engineering department is located on subnet 255.222.43.0. The marketing department is located on 255.222.44.0. You can use this information to control how the engineering and marketing systems are installed with the Solaris 9 software. In the `/jumpstart` directory, you edit the `rules` file, delete all of the example rules, and add the following lines to the file:

```
network 255.222.43.0 - eng_prof -
network 255.222.44.0 - marketing_prof -
```

Basically, these rules state that systems on the 255.222.43.0 network are to be installed with the Solaris 9 software by using the `eng_prof` profile. The systems on the 255.222.44.0 network are to be installed with the Solaris 9 software by using the `marketing_prof` profile.

Note – You can use the sample rules to use a network address to identify the systems to be installed with the Solaris 9 software by using `eng_prof` and `marketing_prof`, respectively. You can also use host names, memory size, or model type as the rule keyword. Table 28–1 contains a complete list of keywords you can use in a `rules` file.

Validate the rules File

After the rules and profiles are set up, you run the check script to verify that the files are correct:

```
server-1# cd /jumpstart
server-1# ./check
```

If the check script does not find any errors, the script creates the `rules.ok` file.

Set Up Engineering Systems to Install From the Network

After setting up the `/jumpstart` directory and files, you use the `add_install_client` command on the install server, `server-1`, to set up the engineering systems to install the Solaris 9 software from the install server. `server-1` is also the boot server for the engineering group's subnet.

```
server-1# cd /export/install/sparc_9/Solaris_9/Tools
server-1# ./add_install_client -c server-1:/jumpstart host-eng1 sun4u
server-1# ./add_install_client -c server-1:/jumpstart host-eng2 sun4u
```

In the `add_install_client` command, the options that are used have the following meanings:

<code>-c</code>	Specifies the server (<code>server-1</code>) and path (<code>/jumpstart</code>) to the JumpStart directory.
<code>host-eng1</code>	The name of a system in the engineering group.
<code>host-eng2</code>	The name of another system in the engineering group.
<code>sun4u</code>	Specifies the platform group of the systems that use <code>server-1</code> as an install server. The platform group is for Ultra 5 systems.

Set Up Marketing Systems to Install From the Network

Next, you use the `add_install_client` command on the boot server (`server-2`) to set up the marketing systems to boot from the boot server and install the Solaris 9 software from the install server (`server-1`):

```
server-2# cd /marketing/boot-dir/Solaris_9/Tools
server-2# ./add_install_client -c server-1:/jumpstart host-mkt1 sun4u
server-2# ./add_install_client -c server-1:/jumpstart host-mkt2 sun4u
```

In the `add_install_client` command, the options that are used have the following meanings:

<code>-c</code>	Specifies the server (<code>server-1</code>) and path (<code>/jumpstart</code>) to the JumpStart directory.
<code>host-mkt1</code>	The name of a system in the marketing group.
<code>host-mkt2</code>	The name of another system in the marketing group.
<code>sun4u</code>	Specifies the platform group of the systems that use <code>server-1</code> as an install server. The platform group is for Ultra 5 systems.

Boot the Systems and Install Solaris 9 Software

After setting up the servers and files, you can boot the systems by using the following boot command at the `ok` (PROM) prompt of each system:

```
ok boot net - install
```

The Solaris operating environment is automatically installed on the systems.

Custom JumpStart (Reference)

This chapter lists keywords and values that you can use in the `rules` file, profiles, and begin and finish scripts.

- “Rule Keywords and Values” on page 189
- “Profile Keywords and Values” on page 194
- “Custom JumpStart Environment Variables” on page 217
- “Probe Keywords and Values” on page 220

Rule Keywords and Values

Table 28–1 describes the keywords and values that you can use in the `rules` file. For detailed instructions to create a `rules` file, see “Creating the `rules` File” on page 147.

TABLE 28–1 Descriptions of Rule Keywords and Values

Keyword	Value	Matches
<code>any</code>	minus sign (-)	Anything. The <code>any</code> keyword always succeeds.
<code>arch</code>	<i>processor_type</i> The valid value for <i>processor_type</i> is <code>sparc</code>	A system’s processor type. The <code>uname -p</code> command reports the system’s processor type.

TABLE 28-1 Descriptions of Rule Keywords and Values (Continued)

Keyword	Value	Matches
disksize	<p><i>actual_disk_name size_range</i></p> <p><i>actual_disk_name</i> – A disk name in the form <i>cxydz</i>, such as <code>c0t3d0</code>, or the special word <code>rootdisk</code>. If <code>rootdisk</code> is used, the disk to be matched is determined in the following order:</p> <ul style="list-style-type: none"> ■ The disk that contains the preinstalled boot image, which is a new system with factory JumpStart installed ■ The <code>c0t3d0s0</code> disk, if the disk exists ■ The first available disk that is searched in kernel probe order <p><i>size_range</i> – The size of the disk, which must be specified as a range of Mbytes (<i>x-x</i>).</p>	<p>The name and size of a system’s disk in Mbytes.</p> <p>Example:</p> <pre>disksize c0t3d0 250-300</pre> <p>In the example, the JumpStart program attempts to match a system disk that is named <code>c0t3d0</code>. The disk can hold between 250 and 300 Mbytes of information.</p> <p>Example:</p> <pre>disksize rootdisk 750-1000</pre> <p>In the example, the JumpStart program first attempts to match a system disk that contains a preinstalled boot image. Next, the JumpStart program attempts to match the <code>c0t3d0s0</code> disk, if the disk exists. Finally, the JumpStart program attempts to match the first available disk that can hold between 750 Mbytes and 1 Gbyte of information.</p> <p>Note – When calculating <i>size_range</i>, remember that a Mbyte equals 1,048,576 bytes. A disk might be advertised as a “535-Mbyte” disk, but the disk might contain only 510 million bytes of disk space. The JumpStart program views the “535-Mbyte” disk as a 510-Mbyte disk because $535,000,000 / 1,048,576 = 510$. A “535-Mbyte” disk does not match a <i>size_range</i> equal to 530–550.</p>
domainname	<i>actual_domain_name</i>	<p>A system’s domain name, which controls how a name service determines information.</p> <p>If you have a system already installed, the <code>domainname</code> command reports the system’s domain name.</p>
hostaddress	<i>actual_IP_address</i>	A system’s IP address.

TABLE 28-1 Descriptions of Rule Keywords and Values (Continued)

Keyword	Value	Matches
hostname	<i>actual_host_name</i>	A system's host name. If you have a system that is already installed, the <code>uname -n</code> command reports the system's host name.
installed	<p><i>slice version</i></p> <p><i>slice</i> — A disk slice name in the form <i>cwtxdysz</i>, such as <code>c0t3d0s5</code>, or the special words <code>any</code> or <code>rootdisk</code>. If <code>any</code> is used, the JumpStart program attempts to match all of the system's disks in kernel probe order. If <code>rootdisk</code> is used, the disk to be matched is determined in the following order:</p> <ul style="list-style-type: none"> ■ The disk that contains the preinstalled boot image, which is a new system with factory JumpStart installed ■ The <code>c0t3d0s0</code> disk, if the disk exists ■ The first available disk that is searched in kernel probe order <p><i>version</i> — A version name or the special words <code>any</code> or <code>upgrade</code>. If <code>any</code> is used, any Solaris or SunOS release is matched. If <code>upgrade</code> is used, any Solaris 2.1 or compatible release that can be upgraded is matched.</p> <p>If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is <code>SystemV</code>.</p>	<p>A disk that has a root (/) file system that corresponds to a particular version of Solaris software.</p> <p>Example:</p> <pre>installed c0t3d0s1 Solaris_9</pre> <p>In the example, the JumpStart program attempts to match a system that has a Solaris 9 root (/) file system on <code>c0t3d0s1</code>.</p>
karch	<p><i>actual_platform_group</i></p> <p>Valid values are <code>sun4m</code>, <code>sun4u</code>, <code>prep</code>. A list of systems and their corresponding platform group is presented in the <i>Solaris 9 Sun Hardware Platform Guide</i>.</p>	<p>A system's platform group.</p> <p>If you have a system that is already installed, the <code>arch -k</code> command or the <code>uname -m</code> command reports the system's platform group.</p>

TABLE 28-1 Descriptions of Rule Keywords and Values (Continued)

Keyword	Value	Matches
memsize	<i>physical_mem</i> The value must be a range of Mbytes , <i>x-x</i> , or a single Mbyte value.	A system's physical memory size in Mbytes. Example: <code>memsize 64-128</code> The example tries to match a system with a physical memory size between 64 and 128 Mbytes. If you have a system that is already installed, the output of the <code>prtconf</code> command, line 2, reports the system's physical memory size.
model	<i>actual_platform_name</i>	A system's platform name. See the <i>Solaris 9 Sun Hardware Platform Guide</i> for a list of valid platform names. To find the platform name of an installed system, use the <code>uname -i</code> command or the output of the <code>prtconf</code> command, line 5. Note – If the <i>actual_platform_name</i> contains spaces, you must replace spaces with underscores (_). Example: <code>SUNW,Sun_4_50</code>
network	<i>network_num</i>	A system's network number, which the JumpStart program determines by performing a logical AND between the system's IP address and the subnet mask. Example: <code>network 193.144.2.8</code> The example tries to match a system with a 193.144.2.8 IP address, if the subnet mask is 255.255.255.0.

TABLE 28-1 Descriptions of Rule Keywords and Values (Continued)

Keyword	Value	Matches
osname	Solaris_x	<p>A version of Solaris software that is already installed on a system.</p> <p>Example:</p> <pre data-bbox="989 516 1195 537">osname Solaris_9</pre> <p>In the example, the JumpStart program attempts to match a system with the Solaris 9 operating environment already installed.</p>
probe	<i>probe_keyword</i>	<p>A valid probe keyword or a valid custom probe keyword.</p> <p>Example:</p> <pre data-bbox="989 814 1130 835">probe disks</pre> <p>The example returns the size of a system's disks in Mbytes and in kernel probe order, for example, c0t3d0s1, c0t4d0s0. The JumpStart program sets the SI_DISKLIST, SI_DISKSIZE, SI_NUMDISKS, and SI_TOTALDISK environment variables.</p> <p>Note – The probe keyword is unique in that the keyword does not attempt to match an attribute and run a profile. The probe keyword returns a value. Consequently, you cannot specify begin scripts, profiles, and finish scripts with the probe rule keyword.</p> <p>Probe keywords are described in Chapter 25.</p>

TABLE 28-1 Descriptions of Rule Keywords and Values (Continued)

Keyword	Value	Matches
totaldisk	<i>size_range</i> The value must be specified as a range of Mbytes (x-x).	<p>The total disk space on a system in Mbytes. The total disk space includes all the operational disks that are attached to a system.</p> <p>Example:</p> <pre>totaldisk 300-500</pre> <p>In the example, the JumpStart program tries to match a system with a total disk space between 300 and 500 Mbytes.</p> <p>Note – When calculating <i>size_range</i>, remember that one Mbyte equals 1,048,576 bytes. A disk might be advertised as a “535-Mbyte” disk, but the disk might have only 510 million bytes of disk space. The JumpStart program views the “535-Mbyte” disk as a 510-Mbyte disk because $535,000,000 / 1,048,576 = 510$. A “535-Mbyte” disk does not match a <i>size_range</i> equal to 530-550.</p>

Profile Keywords and Values

This section describes the profile keywords and values that you can use in a profile. For detailed instructions to create a profile, see “Creating a Profile” on page 150.

Profile Keywords at a Glance

Table 28-2 provides a quick way to determine which keywords you can use, based on your installation scenario. Unless otherwise noted in the keyword descriptions, the keyword can only be used with the initial installation option.

TABLE 28-2 Overview of Profile Keywords

Profile Keywords	Installation Scenarios				
	Standalone System (Non-Networked)	Standalone System (Networked) or Server	OS Server	Upgrade	Upgrade With Disk Space Reallocation
archive_location	✓	✓			
backup_media					✓
boot_device	✓	✓	✓		
client_arch			✓		
client_root			✓		
client_swap			✓		
cluster (adding software groups)	✓	✓	✓		
cluster (adding or deleting clusters)	✓	✓	✓	✓	✓
dontuse	✓	✓	✓		
filesystem (mounting remote file systems)		✓	✓		
filesystem (creating local file systems)	✓	✓	✓		
geo	✓	✓	✓	✓	✓
install_type	✓	✓	✓	✓	✓
isa_bits	✓	✓	✓	✓	✓
layout_constraint					✓
locale	✓	✓	✓	✓	✓
num_clients			✓		
package	✓	✓	✓	✓	✓
partitioning	✓	✓	✓		
root_device	✓	✓	✓	✓	✓
system_type	✓	✓	✓		
usedisk	✓	✓	✓		

Profile Keyword Descriptions and Examples

archive_location Keyword

`archive_location retrieval_type location`

The values of *retrieval_type* and *location* depend on where the Web Start Flash archive is stored. The following sections contain the values you can use for *retrieval_type* and *location* and examples of how to use the `archive_location` keyword.

- “NFS Server” on page 196
- “HTTP Server” on page 196
- “FTP Server” on page 198
- “Local Tape” on page 199
- “Local Device” on page 199
- “Local File” on page 200

NFS Server

If the archive is stored on an NFS server, use the following syntax for the `archive_location` keyword.

```
archive_location nfs server_name:/path/filename retry n
```

<i>server_name</i>	The name of the server where you stored the archive.
<i>path</i>	The location of the archive to be retrieved from the specified server. If the path contains \$HOST, the Web Start Flash installation utilities replace \$HOST with the name of the clone system that you are installing.
<i>filename</i>	The name of the Web Start Flash archive file.
<i>retry n</i>	An optional keyword. <i>n</i> is the maximum number of times the Web Start Flash utilities attempt to mount the archive.

Examples:

```
archive_location nfs golden:/archives/usrarchive
```

```
archive_location nfs://golden/archives/usrarchive
```

HTTP Server

If the archive is stored on an HTTP server, use the following syntax for the `archive_location` keyword.

```
archive_location http://server_name:port path/filename optional_keywords
```

<i>server_name</i>	The name of the server where you stored the archive. <i>server_name</i> can be a port number or the name of a TCP service that has a port number that is determined at runtime.
<i>port</i>	An optional port. If you do not specify a port, the Web Start Flash installation utilities use the default HTTP port number, 80.
<i>path</i>	The location of the archive to be retrieved from the specified server. If the path contains \$HOST, the Web Start Flash installation utilities replace \$HOST with the name of the clone system that you are installing.
<i>filename</i>	The name of the Web Start Flash archive file.
<i>optional_keywords</i>	The optional keywords that you can specify when you retrieve a Web Start Flash archive from an HTTP server.

TABLE 28-3 Optional Keywords to Use With `archive_location http`

Keywords	Value Definitions
<code>auth basic user_name password</code>	<p>If the archive is located on an HTTP server that is password protected, you must include the user name and password that you need to access the HTTP server in the profile file.</p> <p>Note – The use of this authentication method in a profile that is intended for use with custom JumpStart is risky. Unauthorized users might have access to the profile file that contains the password.</p>
<code>timeout min</code>	<p>The <code>timeout</code> keyword enables you to specify, in minutes, the maximum length of time that is allowed to pass without receipt of data from the HTTP server before the connection is closed, reopened, and resumed from the point where the timeout occurred. If you specify a <code>timeout</code> value of 0 (zero), the connection is not reopened because of inactivity.</p> <p>If a time-out reconnection occurs, the Web Start Flash installation utilities attempt to resume the installation at the last known position in the archive. If the Web Start Flash installation utilities cannot resume the installation at the last known position, the retrieval restarts from the beginning of the archive and the data that was retrieved prior to the timeout is discarded.</p>
<code>proxy host:port</code>	<p>The <code>proxy</code> keyword allows you to specify a proxy host and proxy port. You can use a proxy host to retrieve a Web Start Flash archive from the other side of a firewall. You must supply a proxy port when you specify the <code>proxy</code> keyword.</p>

Example:

```
archive_location http://silver/archives/usrarchive.flar timeout 5
```

Example of the `auth basic user_name password` keyword:

```
archive_location http://silver/archives/usrarchive.flar timeout 5 user1 secret
```

FTP Server

If the archive is stored on an FTP server, use the following syntax for the `archive_location` keyword.

```
archive_location ftp://user_name:password@server_name:port path/filename optional_keywords
```

<i>user_name:password</i>	The user name and password that you need to access the FTP server in the profile file.
<i>server_name</i>	The name of the server where you stored the archive. <i>server_name</i> can be a port number or the name of a TCP service that has a port number that is determined at runtime.
<i>port</i>	A is an optional port. If you do not specify a port, the Web Start Flash installation utilities use the default FTP port number, 21.
<i>path</i>	The location of the archive to be retrieved from the specified server. If the path contains \$HOST, the Web Start Flash installation utilities replace \$HOST with the name of the clone system that you are installing.
<i>filename</i>	The name of the Web Start Flash archive file.
<i>optional_keywords</i>	The optional keywords that you can specify when you retrieve a Web Start Flash archive from an FTP server.

TABLE 28-4 Optional Keywords to Use With `archive_location ftp`

Keywords	Value Definitions
<code>timeout min</code>	<p>The <code>timeout</code> keyword enables you to specify, in minutes, the maximum length of time that is allowed to pass without receipt of data from the FTP server before the connection is closed, reopened, and resumed from the point where the timeout occurred. If you specify a <code>timeout</code> value of 0 (zero), the connection is not reopened because of inactivity.</p> <p>If a time-out reconnection occurs, the Web Start Flash installation utilities attempt to resume the installation at the last known position in the archive. If the Web Start Flash installation utilities cannot resume the installation at the last known position, the retrieval restarts from the beginning of the archive and the data that was retrieved prior to the timeout is discarded.</p>
<code>proxy host:port</code>	The <code>proxy</code> keyword allows you to specify a proxy host and proxy port. You can use a proxy host to retrieve a Web Start Flash archive from the other side of a firewall. You must supply a proxy port when you specify the <code>proxy</code> keyword.

Example:

```
archive_location ftp://user1:secret@silver/archives/usrarchive.flar timeout 5
```

Local Tape

If the archive is stored on a tape, use the following syntax for the `archive_location` keyword.

```
archive_location local_tape device position
```

<i>device</i>	The name of the tape drive where you stored the Web Start Flash archive. If the device name is a canonical path, the Web Start Flash installation utilities retrieve the archive from the path to the device node. If you supply a device name that is not a canonical path, the Web Start Flash installation utilities add <code>/dev/rmt/</code> to the path.
<i>position</i>	Designates the place on the tape drive where you saved the archive. If you do not supply a position, the Web Start Flash installation utilities retrieve the archive from the current position on the tape drive. By specifying a <i>position</i> , you can place a begin script or a <code>sysidcfg</code> file on the tape drive before the archive.

Examples:

```
archive_location local_tape /dev/rmt/0n 5
```

```
archive_location local_tape 0n 5
```

Local Device

You can retrieve a Web Start Flash archive from a local device if you stored the Web Start Flash archive on a file system-oriented, random-access device, such as a diskette or a CD-ROM. Use the following syntax for the `archive_location` keyword.

Note – You can retrieve an archive from stream-oriented devices, such as tape, by using the syntax for local tape.

```
archive_location local_device device path/filename file_system_type
```

<i>device</i>	The name of the drive where you stored the Web Start Flash archive. If the device name is a canonical path, the device is mounted directly. If you supply a device name that is not a canonical path, the Web Start Flash installation utilities add <code>/dev/dsk/</code> to the path.
<i>path</i>	The path to the Web Start Flash archive, relative to the root of the file system on the device you specified. If the path contains <code>\$HOST</code> , the Web Start Flash installation utilities replace <code>\$HOST</code> with the name of the clone system that you are installing.

<i>filename</i>	The name of the Web Start Flash archive file.
<i>file_system_type</i>	Specifies the type of file system on the device. If you do not supply a file system type, the Web Start Flash installation utilities attempt to mount a UFS file system. If the UFS mount fails, the Web Start Flash installation utilities attempt to mount an HSFS file system.

Examples:

To retrieve an archive from a local hard drive that is formatted as a UFS file system, use the following command:

```
archive_location local_device c0t0d0s0 /archives/$HOST
```

To retrieve an archive from a local CD-ROM that has an HSFS file system, use the following command:

```
archive_location local_device c0t0d0s0 /archives/usrarchive
```

Local File

You can retrieve an archive that you stored in the miniroot from which you booted the clone system as a local file. When you perform a custom JumpStart installation, you boot the system from a CD-ROM or an NFS-based miniroot. The installation software is loaded and run from this miniroot. Therefore, a Web Start Flash archive that you stored in the CD-ROM or NFS-based miniroot is accessible as a local file. Use the following syntax for the `archive_location` keyword.

```
archive_location local_file path/filename
```

<i>path</i>	The location of the archive. The path must be accessible to the system as a local file while the system is booted from the Solaris 9 Installation CD or from the Solaris 9 DVD. The system cannot access <code>/net</code> when it is booted from the Solaris 9 Installation CD or from the Solaris 9 DVD.
-------------	--

<i>filename</i>	The name of the Web Start Flash archive file.
-----------------	---

Examples:

```
archive_location local_file /archives/usrarchive
```

```
archive_location local_file /archives/usrarchive
```

backup_media Profile Keyword

```
backup_media type path
```

Note – You can use `backup_media` only with the upgrade option when disk space reallocation is required.

`backup_media` defines the media that is to be used to back up file systems if space needs to be reallocated during an upgrade because of insufficient space. If multiple tapes or diskettes are required for the backup, you are prompted to insert tapes or diskettes during the upgrade.

Valid <i>type</i> Values	Valid <i>path</i> Values	Specification
local_tape	<code>/dev/rmt/n</code>	A local tape drive on the system that is being upgraded. <i>path</i> must be the character (raw) device path for the tape drive. <i>n</i> is the number of the tape drive.
local_diskette	<code>/dev/rdisketten</code>	A local diskette drive on the system that is being upgraded. <i>path</i> must be the character (raw) device path for the diskette drive. <i>n</i> is the number of the diskette drive. Diskettes that you use for the backup must be formatted.
local_filesystem	<code>/dev/dsk/cwtxdysz</code> <code>/file_system</code>	A local file system on the system that is being upgraded. You cannot specify a local file system that is being changed by the upgrade. <i>path</i> can be a block device path for a disk slice. For example, the <code>tx</code> in <code>/dev/dsk/cwtxdysz</code> might not be needed. Or, <i>path</i> can be the absolute path to a file system that is mounted by the <code>/etc/vfstab</code> file.
remote_filesystem	<code>host : /file_system</code>	An NFS file system on a remote system. <i>path</i> must include the name or IP address of the remote system, <i>host</i> , and the absolute path to the NFS file system, <i>file_system</i> . The NFS file system must have read/write access.
remote_system	<code>user@host : /directory</code>	A directory on a remote system that can be reached by a remote shell, <code>rsh</code> . The system that is being upgraded must have access to the remote system through the remote system's <code>.rhosts</code> file. <i>path</i> must include the name of the remote system <i>host</i> and the absolute path to the directory <i>directory</i> . If a user login ID <i>user</i> is not specified, <code>root</code> is used by default.

Examples:

```
backup_media local_tape /dev/rmt/0

backup_media local_diskette /dev/rdiskette1

backup_media local_filesystem /dev/dsk/c0t3d0s4

backup_media local_filesystem /export

backup_media remote_filesystem system1:/export/temp

backup_media remote_system user1@system1:/export/temp
```

boot_device Profile Keyword

`boot_device device eeprom`

`boot_device` designates the device where the JumpStart program is to install the root (/) file system and the system's boot device.

If you do not specify the `boot_device` keyword in a profile, the following `boot_device` keyword is specified by default during the installation: `boot_device any update`.

device – Use one of the following values.

- *cwtxdysz* or *cxdysz* – The disk slice where the JumpStart program places the root (/) file system, for example, `c0t0d0s0`.
- *existing* – The JumpStart program places the root (/) file system on the system's existing boot device.
- *any* – The JumpStart program chooses where to place the root (/) file system. The JumpStart program attempts to use the system's existing boot device. The JumpStart program might choose a different boot device if necessary.

eeprom – Choose to update or preserve the system's EEPROM.

Choose if you want to update or preserve the system's EEPROM to the specified boot device.

- *update* – The JumpStart program updates the system's EEPROM to the specified boot device so that the installed system automatically boots from it.
- *preserve* – The boot device value in the system's EEPROM is not changed. If you specify a new boot device without changing the system's EEPROM, you need to change the system's EEPROM manually so it can automatically boot from the new boot device.

Note – The *eeprom* value also allows you to update the system’s EEPROM if you change the system’s current boot device. By updating the system’s EEPROM, the system can automatically boot from the new boot device.

Example:

```
boot_device c0t0d0s2 update
```

Note – *boot_device* must match any *filesystem* keywords that specify the root (/) file system and the *root_device* keyword, if specified.

client_arch Profile Keyword

```
client_arch karch_value ...
```

client_arch specifies that the operating system server is to support a different platform group than the server uses. If you do not specify *client_arch* in the profile, any diskless client that uses the operating system server must contain the same platform group as the server. You must specify each platform group that you want the operating system server to support.

Valid values for *karch_value* are *sun4m* and *sun4u*. For a detailed list of platform names and various systems, see *Solaris 9 Sun Hardware Guide*.

Note – You can use *client_arch* only when *system_type* is specified as *server*.

client_root Profile Keyword

```
client_root root_size
```

client_root defines the amount of root space, *root_size* in Mbytes, to allocate for each client. If you do not specify *client_root* in a server’s profile, the installation software allocates 15 Mbytes of root space per client. The size of the client root area is used in combination with the *num_clients* keyword to determine how much space to reserve for the */export/root* file system.

Note – You can use *client_root* only when *system_type* is specified as *server*.

client_swap Profile Keyword

`client_swap swap_size`

`client_swap` defines the amount of swap space, *swap_size* in Mbytes, to allocate for each diskless client. If you do not specify `client_swap` in the profile, 32 Mbytes of swap space is allocated by default.

Example:

```
client_swap 64
```

The example specifies that each diskless client is to have a swap space of 64 Mbytes.

Note – You can use `client_swap` only when `system_type` is specified as `server`.

How the Size of swap Is Determined

If a profile does not specify the size of swap, the JumpStart program determines the size of the swap space, based on the system's physical memory. Table 28–5 shows how the size of swap is determined during a custom JumpStart installation.

TABLE 28–5 Determining swap Size

Physical Memory (in Mbytes)	Swap Space (in Mbytes)
16–64	32
64–128	64
128–512	128
Greater than 512	256

The JumpStart program makes the size of swap no more than 20 percent of the disk where swap is located, unless the disk contains free space after laying out the other file systems. If free space exists, the JumpStart program allocates the free space to swap, and if possible, allocates the amount that is shown in Table 28–5.

Note – Physical memory plus swap space must total a minimum of 32 Mbytes.

cluster Profile Keyword (Adding Software Groups)

`cluster group_name`

`cluster` designates the software group to add to the system. The *group_name* for each software group is listed in the following table.

Software Group	group_name
Core	SUNWCreq
End User Solaris Software Group	SUNWCuser
Developer Solaris Software Group	SUNWCprog
Entire Solaris Software Group	SUNWCall
Entire Solaris Software Group Plus OEM Support	SUNWCXall

You can specify only one software group in a profile. The software group must be specified before other `cluster` and `package` entries. If you do not specify a software group with `cluster` in the profile, the end-user software group, `SUNWCuser`, is installed on the system.

cluster Profile Keyword (Adding or Deleting Clusters)

`cluster cluster_name add_delete_switch`

Note – `cluster` (adding or deleting clusters) can be used with both the initial installation and upgrade options.

`cluster` designates whether a cluster is to be added or deleted from the software group that is to be installed on the system.

`cluster_name` must be in the form `SUNWCname`. To view detailed information about clusters and their names, start `Admintool` on an installed system and choose `Software` from the `Browse` menu.

`add_delete_switch` represents the option `add` or `delete`. Use `add_delete_switch` to indicate whether to add or delete the cluster that is specified. If you do not specify `add_delete_switch`, `add` is used by default.

When you use `cluster` (adding or deleting clusters) during an upgrade, the following conditions apply:

- All clusters that are already on the system are automatically upgraded.
- If you specify `cluster_name add`, and `cluster_name` is not installed on the system, the cluster is installed.
- If you specify `cluster_name delete`, and `cluster_name` is installed on the system, the package is deleted *before* the upgrade begins.

dontuse Profile Keyword

`dontuse disk_name ...`

By default, the JumpStart program uses all of the operational disks on the system when `partitioning default` is specified. `dontuse` designates one or more disks that you do not want the JumpStart program to use. `disk_name` must be specified in the form `cxydz`, for example, `c0t0d0`.

Note – You cannot specify the `dontuse` keyword and the `usedisk` keyword in the same profile.

filesystem Profile Keyword (Mounting Remote File Systems)

`filesystem server:path server_address mount_pt_name [mount_options]`

By using `filesystem` with the listed values, the JumpStart program sets up the installed system to automatically mount remote file systems when the system boots. You can specify `filesystem` more than once.

Example:

```
filesystem sherlock:/export/home/user2 - /home
```

server: – The name of the server where the remote file system is located, followed by a colon.

path – The remote file system's mount-point name. For example, `/usr` or `/export/home`.

server_address – The IP address of the server that is specified in *server:path*. If a name service is not running on the network, the *server_address* value can be used to populate the `/etc/hosts` file with the server's host name and IP address. If you are not specifying the server's IP address, you must specify a minus sign (-). For example, if you have a name service that is running on the network, you do not need to specify the server's IP address.

mount_pt_name – The name of the mount point on which the remote file system is to be mounted.

mount_options – One or more mount options, which is the same as the `-o` option of the `mount(1M)` command. The mount options are added to the `/etc/vfstab` entry for the specified *mount_pt_name*.

Note – If you need to specify more than one mount option, the mount options must be separated by commas and no spaces (`ro,quota` for example).

fileSYS Profile Keyword (Creating Local File Systems)

`fileSYS slice size [file_system optional_parameters]`

By using `fileSYS` with the values that are listed, the JumpStart program creates local file systems during the installation. You can specify `fileSYS` more than once.

slice – Use one of the following values:

- `any` – The JumpStart program places the file system on any disk.

Note – You cannot specify `any` when `size` is `existing`, `all`, `free`, `start:size`, or `ignore`.

- `cwtXdysz` – The disk slice where the JumpStart program places the file system, for example, `c0t0d0s0` or `c0d0s0`.
- `rootdisk.sn` – The variable that contains the value for the system’s root disk, which is determined by the JumpStart program as described in “How the System’s Root Disk Is Determined” on page 215. The `sn` suffix indicates a specific slice on the disk.

size – Use one of the following values:

- `num` – The size of the file system is set to `num` in Mbytes.
- `existing` – The current size of the existing file system is used.

Note – When you use the `existing` value, you can change the name of an existing slice by specifying `file_system` as a different `mount_pt_name`.

- `auto` – The size of the file system is automatically determined, depending on the software that is selected.
- `all` – The specified *slice* uses the entire disk for the file system. When you specify the `all` value, no other file systems can be placed on the specified disk.
- `free` – The remaining unused space on the disk is used for the file system.

Note – If `free` is used as the value to `fileSYS`, the `fileSYS` entry must be the last entry in a profile.

- *start:size* – The file system is explicitly partitioned. *start* is the cylinder where the slice begins. *size* is the number of cylinders for the slice.

file_system – The *file_system* value is optional and used when *slice* is specified as any or *cwtxdysz*. If *file_system* is not specified, `unnamed` is set by default. If `unnamed` is set, you cannot specify the *optional_parameters* value. Use one of the following values:

- *mount_pt_name* – The file system's mount-point name, for example, `/var`.
- `swap` – The specified *slice* is used as swap.
- `overlap` – The specified *slice* is defined as a representation of a disk region. The VTOC value is `V_BACKUP`. By default, slice 2 is an overlap slice that is a representation of the whole disk.

Note – You can specify `overlap` only when *size* is `existing`, `all`, or *start:size*.

- `unnamed` – The specified *slice* is defined as a raw slice, so *slice* does not have a mount-point name. If you do not specify *file_system*, `unnamed` is used by default.
- `ignore` – The specified *slice* is not used or recognized by the JumpStart program. You can use this option to specify that you want a file system to be ignored on a disk during installation. The JumpStart program creates a new file system on the same disk with the same name. You can use `ignore` only when `partitioning existing` is specified.

optional_parameters – Use one of the following values:

- `preserve` – The file system on the specified *slice* is preserved.

Note – `preserve` can be specified only when *size* is `existing` and *slice* is *cwtxdysz*.

- *mount_options* – One or more mount options, which is the same as the `-o` option of the `mount(1M)` command. The mount options are added to the `/etc/vfstab` entry for the specified *mount_pt_name*.

Note – If you need to specify more than one mount option, the mount options must be separated by commas and no spaces, for example, `ro,quota`, for example.

geo Profile Keyword

geo locale

Note – You can use `geo` with both the initial installation and upgrade options.

`geo` designates the regional locale or locales that you want to install on a system or to add when upgrading a system. Values you can specify for *locale* are listed in the following table.

Value	Description
N_Africa	Northern Africa, including Egypt
C_America	Central America, including Costa Rica, El Salvador, Guatemala, Mexico, Nicaragua, Panama
N_America	North America, including Canada, United States
S_America	South America, including Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Paraguay, Peru, Uruguay, Venezuela
Asia	Asia, including Japan, Republic of Korea, People's Republic of China, Taiwan, Thailand
Ausi	Australasia, including Australia, New Zealand
C_Europe	Central Europe, including Austria, Czech Republic, Germany, Hungary, Poland, Slovakia, Switzerland
E_Europe	Eastern Europe, including Albania, Bosnia, Bulgaria, Croatia, Estonia, Latvia, Lithuania, Macedonia, Romania, Russia, Serbia, Slovenia, Turkey
N_Europe	Northern Europe, including Denmark, Finland, Iceland, Norway, Sweden
S_Europe	Southern Europe, including Greece, Italy, Portugal, Spain
W_Europe	Western Europe, including Belgium, France, Great Britain, Ireland, Netherlands

Value	Description
M_East	Middle East, including Israel

A complete list of the component locale values that compose each regional locale that is listed previously is presented in Chapter 40.

Note – You can specify a geo keyword for each locale you need to add to a system.

install_type Profile Keyword

`install_type` *initial_upgrade_flash_switch*

`install_type` defines whether to erase and install a new Solaris operating environment on a system, upgrade the existing Solaris environment on a system, or install a Web Start Flash archive on the system.

Note – You must specify `install_type` in a profile, and `install_type` must be the first profile keyword in every profile.

You must use one of the following options for the *initial_upgrade_flash_switch*:

- `initial_install` – Specifies to perform an initial installation of the Solaris operating environment
- `upgrade` – Specifies to perform an upgrade of the Solaris operating environment
- `flash_install` – Specifies to install a Web Start Flash archive

Note – Some profile keywords can only be used with the `initial_install` option. Some profile keywords can only be used with the `upgrade` option. Some profile keywords can only be used with the `flash_install` option.

isa_bits Profile Keyword

`isa_bits` *bit_switch*

`isa_bits` specifies whether 64-bit or 32-bit Solaris 9 packages are to be installed.

bit_switch represents the option 64 or 32, which you use to indicate whether 64-bit or 32-bit Solaris 9 packages are to be installed. If you do not set this keyword in the profile, the JumpStart program installs systems as follows:

- 64-bit packages on UltraSPARC™ systems
- 32-bit packages on all other systems

Note – If you use the `isa_bits` keyword, you must also use the latest check script in the `solaris_9/Misc/jumpstart_sample` directory on the Solaris 9 Software 1 of 2 CD or on the Solaris 9 DVD.

layout_constraint Profile Keyword

`layout_constraint slice constraint [minimum_size]`

Note – You can use `layout_constraint` only for the upgrade option when you need to reallocate disk space.

`layout_constraint` designates the constraint auto-layout has on a file system if auto-layout needs to reallocate space during an upgrade because of space problems.

If you do not specify the `layout_constraint` keyword, the JumpStart program lays out the disk as follows:

- File systems that require more space for the upgrade are marked changeable.
- File systems that are on the same disk as the file system that requires more space and that are mounted by the `/etc/vfstab` file are marked changeable.
- Remaining file systems are marked fixed because auto-layout cannot change the file systems.

If you specify one or more `layout_constraint` keywords, the JumpStart program lays out the disk as follows:

- File systems that require more space for the upgrade are marked changeable.
- File systems for which you specified a `layout_constraint` keyword are marked with the specified constraint.
- The remaining file systems are marked fixed.

You cannot change the constraint on file systems that require more space for the upgrade because the file systems must be marked changeable. You can use the `layout_constraint` keyword to change the `minimum_size` values on file systems that require more space for the upgrade.

Note – To help auto-layout reallocate space, select more file systems to be changeable or movable, especially those file systems that are located on the same disks as the file systems that require more space for the upgrade.

slice – *slice* specifies the file system's disk slice on which to specify the constraint. You must specify the system's disk slice in the form *cwt.xdysz*.

constraint – Use one of the following constraints for the specified file system:

- *changeable* – Auto-layout can move the file system to another location and it can change the file system size. The *changeable* constraint can only be specified on file systems that are mounted by the */etc/vfstab* file. You can change the file system's size by specifying the *minimum_size* value.

When you mark a file system as changeable and *minimum_size* is not specified, the file system's minimum size is set to 10 percent more than the minimum size that is required. For example, if the minimum size for a file system is 100 Mbytes, the changed size is 110 Mbytes. If *minimum_size* is specified, any free space that remains, original size minus minimum size, is used for other file systems.

- *movable* – Auto-layout can move the file system to another slice on the same disk or different disk. The file system size remains the same.
- *available* – Auto-layout can use all of the space on the file system to reallocate space. All of the data in the file system is lost. The *available* constraint can only be specified on file systems that are not mounted by the */etc/vfstab* file.
- *collapse* – Auto-layout moves and collapses the specified file system into the parent file system. You can use the *collapse* option to reduce the number of file systems on a system as part of the upgrade. For example, if a system has the */usr* and */usr/share* file systems, collapsing the */usr/share* file system moves the file system into */usr*, the parent file system. You can specify the *collapse* constraint only on file systems that are mounted by the */etc/vfstab* file.

minimum_size – Specifies the size of the file system after auto-layout reallocates space. The *minimum_size* option enables you to change the size of a file system. The size of the file system might be larger if unallocated space is added to the file system. But, the size is never less than the value you specify. The *minimum_size* value is optional. Use this value only if you have marked a file system as changeable and the minimum size cannot be less than what the file system needs for the existing file system contents.

Examples:

```
layout_constraint c0t3d0s1 changeable 200
```

```
layout_constraint c0t3d0s4 movable
```

```
layout_constraint c0t3d1s3 available
```

```
layout_constraint c0t2d0s1 collapse
```

locale Profile Keyword

`locale locale_name`

Note – You can use `locale` with both the initial installation and upgrade options.

`locale` designates the locale packages you want to install or add when upgrading for the specified `locale_name`. The `locale_name` values are the same as those values that are used for the `$LANG` environment variable. Chapter 40 contains a list of valid locale values.

When you use the `local` keyword, consider the following:

- If you have preconfigured a default locale, the locale is automatically installed. The English language packages are installed by default.
- You can specify a `locale` keyword for each locale you need to add to a system.

num_clients Profile Keyword

`num_clients client_num`

When a server is installed, space is allocated for each diskless client's root (/) and swap file systems. `num_clients` defines the number of diskless clients, `client_num`, that a server supports. If you do not specify `num_clients` in the profile, five diskless clients are allocated by default.

Note – You can use `num_clients` only when `system_type` is specified as `server`.

package Profile Keyword

`package package_name [add_delete_switch]`

Note – You can use `package` with both the initial installation and upgrade options.

`package` designates whether a package is to be added to or deleted from the software group that is to be installed on the system.

You must specify `package_name` in the form `SUNWname`. To view detailed information about packages and their names, on an installed system, use the `pkginfo -l` command.

add_delete_switch represents the option `add` or `delete`, which you use to indicate whether to add or delete the specified package. If you do not specify *add_delete_switch*, `add` is used by default.

Note – Some packages are required and cannot be deleted. You cannot individually add or delete localization packages by using the `package` profile keyword. To add localization packages, use the `locale` profile keyword.

When you use `package` for an upgrade, the JumpStart program performs the following actions:

- All packages already on the system are automatically upgraded.
- If you specify *package_name* `add` and *package_name* is not installed on the system, the package is installed.
- If you specify *package_name* `delete` and *package_name* is installed on the system, the package is deleted *before* the upgrade begins.
- If you specify *package_name* `delete` and *package_name* is not installed on the system, the package is not installed if the package is part of a cluster that is designated to be installed.

partitioning Profile Keyword

partitioning type

partitioning defines how the disks are divided into slices for file systems during the installation.

type – Use one of the following values:

- `default` – The JumpStart program selects the disks and creates the file systems on which to install the specified software, except for any file systems that are specified by the `filesys` keywords. `rootdisk` is selected first. The JumpStart program uses additional disks if the specified software does not fit on `rootdisk`.
- `existing` – The JumpStart program uses the existing file systems on the system's disks. All file systems except `/`, `/usr`, `/usr/openwin`, `/opt`, and `/var` are preserved. The JumpStart program uses the last mount-point field from the file system superblock to determine which file system mount point the slice represents.

Note – When you use both the `filesys` and `partitioning existing` profile keywords, you must set *size size* to `existing`.

- `explicit` – The JumpStart program uses the disks and creates the file systems that are specified by the `filesys` keywords. If you specify only the root (/) file system with the `filesys` keyword, all of the Solaris software is installed in the root (/) file system.

Note – If you use the `explicit` profile value, you must use the `filesys` keyword to specify the disks to use and file systems to create.

If you do not specify partitioning in the profile, the default type of partitioning is used by default.

root_device Profile Keyword

`root_device slice`

Note – You can use `root_device` with both the initial installation and upgrade options.

`root_device` designates the system's root disk. "How the System's Root Disk Is Determined" on page 215 contains additional information.

When you are upgrading a system, `root_device` designates the root (/) file system and the file systems that are mounted by its `/etc/vfstab` file to be upgraded. You must specify `root_device` if more than one root (/) file system can be upgraded on a system. You must specify `slice` in the form `cwtxdysz`.

Example:

```
root_device c0t0d0s2
```

When you use the `root_device` keyword, consider the following:

- If you specify `root_device` on a system with only one disk, the `root_device` and the disk must match. Also, any `filesys` keywords that specify the root (/) file system must match `root_device`.
- If you are upgrading a mirror, the value that is specified for `root_device` should be one side of the mirror. The other side of the mirror is automatically upgraded.

How the System's Root Disk Is Determined

A system's root disk is the disk on the system that contains the root (/) file system. In a profile, you can use the `rootdisk` variable in place of a disk name, which the JumpStart program sets to the system's root disk. Table 28-6 describes how the JumpStart program determines the system's root disk for the installation.

Note – The JumpStart program only determines a system’s root disk size during an initial installation. You cannot change a system’s root disk during an upgrade.

TABLE 28-6 How JumpStart Determines a System’s Root Disk (Initial Installation)

Stage	Action
1	If the <code>root_device</code> keyword is specified in the profile, the JumpStart program sets <code>rootdisk</code> to the root device.
2	If <code>rootdisk</code> is not set and the <code>boot_device</code> keyword is specified in the profile, the JumpStart program sets <code>rootdisk</code> to the boot device.
3	If <code>rootdisk</code> is not set and a <code>filesys cwtxdysz size /</code> entry is specified in the profile, the JumpStart program sets <code>rootdisk</code> to the disk that is specified in the entry.
4	If <code>rootdisk</code> is not set and a <code>rootdisk.sn</code> entry is specified in the profile, the JumpStart program searches the system’s disks in kernel probe order for an existing root file system on the specified slice. If a disk is found, the JumpStart program sets <code>rootdisk</code> to the found disk.
5	If <code>rootdisk</code> is not set and <code>partitioning existing</code> is specified in the profile, the JumpStart program searches the system’s disks in kernel probe order for an existing root file system. If a root file system is not found or more than one is found, an error occurs. If a root file system is found, the JumpStart program sets <code>rootdisk</code> to the found disk.
6	If <code>rootdisk</code> is not set, the JumpStart program sets <code>rootdisk</code> to the disk where the root (/) file system is installed.

system_type Profile Keyword

`system_type type_switch`

`system_type` defines the type of system on which the Solaris environment is to be installed.

`type_switch` represents the option `standalone` or `server`, which you use to indicate the type of system on which the Solaris software is to be installed. If you do not specify `system_type` in a profile, `standalone` is used by default.

usedisk Profile Keyword

`usedisk disk_name ...`

By default, the JumpStart program uses all of the operational disks on the system when you specify `partitioning default`. The `usedisk` profile keyword designates one or more disks that you want the JumpStart program to use. You must specify `disk_name` in the form `cxyzdz` or `cydz`, for example, `c0t0d0` or `c0d0s0`.

If you specify `usedisk` in a profile, the JumpStart program uses only the disks that you specify after the `usedisk` keyword.

Note – You cannot specify the `usedisk` keyword and the `dontuse` keyword in the same profile.

Custom JumpStart Environment Variables

You can use environment variables in your begin and finish scripts. For example, a begin script might extract the disk size, `SI_DISKSIZES`, and install or not install particular packages on a system, based on the actual disk size the script extracts.

Information that is gathered about a system is stored in these environment variables, which are generally set or not, depending on the rule keywords and values you use in the `rules` file.

For example, information about which operating system is already installed on a system is only available in `SI_INSTALLED` after the `installed` keyword is used.

Table 28–7 describes these variables and their values.

TABLE 28–7 Installation Environment Variables

Environment Variable	Value
<code>CHECK_INPUT</code>	The path to the <code>rules</code> file in the JumpStart directory, which is mounted on <code>/tmp/install_config/rules</code> .
<code>HOME</code>	The root's home directory during installation, which is <code>/tmp/root</code> .
<code>PATH</code>	The shell search path during installation, which is <code>/sbin:/usr/sbin/install.d:/usr:/usr/sbin:/usr/bin</code> .
<code>SI_ARCH</code>	The hardware architecture of the install client. The <code>SI_ARCH</code> variable is set when the <code>arch</code> keyword is used in the <code>rules</code> file.
<code>SI_BEGIN</code>	The name of the begin script, if one is used.

TABLE 28-7 Installation Environment Variables (Continued)

Environment Variable	Value
SI_CLASS	The name of the profile that is used to install the install client.
SI_CONFIG_DIR	The path to the JumpStart directory, which is mounted on /tmp/instal_config.
SI_CONFIG_FILE	The path to the rules file in the JumpStart directory, which is mounted on /tmp/install_config/rules.
SI_CONFIG_PROG	The rules file.
SI_CUSTOM_PROBES_FILE	The custom_probes.ok file, in which you can define your own rule and probe keywords. If you create a custom_probes.ok file, you can use the file to extend the list of default rule keywords, which are described in "Rule Keywords and Values" on page 189. You can also use the file to extend the list of default probe keywords that are described in "Probe Keywords and Values" on page 220.
SI_DISKLIST	A comma-separated list of disk names on the install client. The SI_DISKLIST variable is set when the disksize keyword is used and matched in the rules file. The SI_DISKLIST and SI_NUMDISKS variables are used to determine the physical disk to use for the rootdisk. rootdisk is described in "How the System's Root Disk Is Determined" on page 215.
SI_DISKSIZE	A comma-separated list of disk sizes on the install client. The SI_DISKSIZE variable is set when the disksize keyword is used and matched in the rules file.
SI_DOMAINNAME	The domain name. The SI_DOMAINNAME variable is set when the domainname keyword is used and matched in the rules file.
SI_FINISH	The name of the finish script, if one is used.
SI_HOSTADDRESS	The install client's IP address.
SI_HOSTID	The install client's Ethernet address.
SI_HOSTNAME	The install client's host name. The SI_HOSTNAME variable is set when the hostname keyword is used and matched in the rules file.
SI_INSTALLED	The device name of a disk with a specific operating system on the disk, for example, Solaris, SunOS, or System V. The SI_INSTALLED variable is set when the installed keyword is used and matched in the rules file. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.
SI_INST_OS	The name of the operating system. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.
SI_INST_VER	The version of the operating system. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.
SI_KARCH	The install client's kernel architecture. The SI_KARCH variable is set when the karch keyword is used and matched in the rules file.

TABLE 28-7 Installation Environment Variables (Continued)

Environment Variable	Value
SI_MEMSIZE	The amount of physical memory on the install client. The <code>SI_MEMSIZE</code> variable is set when the <code>memsize</code> keyword is used and matched in the <code>rules</code> file.
SI_MODEL	The install client's model name. The <code>SI_MODEL</code> variable is set when the <code>model</code> keyword is used and matched in the <code>rules</code> file.
SI_NETWORK	The install client's network number. The <code>SI_NETWORK</code> variable is set when the <code>network</code> keyword is used and matched in the <code>rules</code> file.
SI_NUMDISKS	The number of disks on an install client. The <code>SI_NUMDISKS</code> variable is set when the <code>disksize</code> keyword is used and matched in the <code>rules</code> file. The <code>SI_NUMDISKS</code> and <code>SI_DISKLIST</code> variables are used to determine the physical disk to use for the <code>rootdisk</code> . <code>rootdisk</code> is described in "How the System's Root Disk Is Determined" on page 215.
SI_OSNAME	The operating system release on the Solaris 9 software image. For example, you can use the <code>SI_OSNAME</code> variable in a script if you are installing the Solaris software on systems that are based on the version of the operating system on the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD image.
SI_PROFILE	The path to the profile in the mounted JumpStart directory. The path is <code>/tmp/install_config/profile_name</code> . If you are creating a derived profile, <code>SI_PROFILE</code> is set to the <code>/tmp/install.input</code> file.
SI_ROOTDISK	The device name of the disk that is represented by the logical name <code>rootdisk</code> . The <code>SI_ROOTDISK</code> variable is set when the <code>disksize</code> or the <code>installed</code> keyword is set to <code>rootdisk</code> in the <code>rules</code> file.
SI_ROOTDISKSIZE	The size of the disk that is represented by the logical name <code>rootdisk</code> . The <code>SI_ROOTDISKSIZE</code> variable is set when the <code>disksize</code> or the <code>installed</code> keyword is set to <code>rootdisk</code> in the <code>rules</code> file.
SI_SYS_STATE	The <code>/a/etc/.sysidtool.state</code> file. You can edit this file in a finish script to prevent the <code>sysidroot</code> program from prompting for a root password before the system reboots.
SI_TOTALDISK	The total amount of disk space on the install client. The <code>SI_TOTALDISK</code> variable is set when the <code>totaldisk</code> keyword is used and matched in the <code>rules</code> file.
SHELL	The default shell during installation, which is <code>/sbin/sh</code> .
TERM	The install client's terminal type.
TZ	The default time zone, as specified in the NIS or NIS+ name service.

Probe Keywords and Values

Table 28–8 describes each rule keyword and its equivalent probe keyword.

Note – Always place probe keywords at or near the beginning of the `rules` file.

TABLE 28–8 Descriptions of Probe Keywords

Rule Keyword	Equivalent Probe Keyword	Description of Probe Keyword
any	None	
arch	arch	Determines the kernel architecture, SPARC, and sets <code>SI_ARCH</code> .
disksize	disks	Returns the size of a system's disks in Mbytes in kernel probe order, <code>c0t3d0s0</code> , <code>c0t3d0s1</code> , <code>c0t4d0s0</code> . <code>disksize</code> sets <code>SI_DISKLIST</code> , <code>SI_DISKIZES</code> , <code>SI_NUMDISKS</code> , and <code>SI_TOTALDISK</code> .
domainname	domainname	Returns a system's NIS or NIS+ domain name or blank and sets <code>SI_DOMAINNAME</code> . The <code>domainname</code> keyword returns the output of <code>domainname(1M)</code> .
hostaddress	hostaddress	Returns a system's IP address, the first address that is listed in the output of <code>ifconfig(1M) -a</code> that is not <code>lo0</code> , and sets <code>SI_HOSTADDRESS</code> .
hostname	hostname	Returns a system's host name that is the output from <code>uname(1) -n</code> and sets <code>SI_HOSTNAME</code> .
installed	installed	Returns the version name of the Solaris operating environment that is installed on a system and sets <code>SI_ROOTDISK</code> and <code>SI_INSTALLED</code> . If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is <code>SystemV</code> .
karch	karch	Returns a system's platform group, for example <code>sun4m</code> , and <code>sun4</code> , and sets <code>SI_KARCH</code> . For a list of platform names, see <i>Solaris 9 Sun Hardware Platform Guide</i> .
memsize	memsize	Returns the size of physical memory on a system in Mbytes and sets <code>SI_MEMSIZE</code> .
model	model	Returns a system's platform name and sets <code>SI_MODEL</code> . For a list of platform names, see the <i>Solaris 9 Sun Hardware Platform Guide</i> .

TABLE 28-8 Descriptions of Probe Keywords (Continued)

Rule Keyword	Equivalent Probe Keyword	Description of Probe Keyword
network	network	Returns a system's network number, which the JumpStart program determines by performing a logical AND between the system's IP address and the subnet mask. The system's IP address and the subnet mask are extracted from the first address that is listed in the output of <code>ifconfig(1M) -a</code> that is not <code>lo0</code> . The <code>network</code> keyword sets <code>SI_NETWORK</code> .
osname	osname	Returns the version and operating system name of the Solaris operating environment that is found on a CD and sets <code>SI_OSNAME</code> . If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is <code>SystemV</code> .
	rootdisk	Returns the name and size in Mbytes of a system's root disk and sets <code>SI_ROOTDISK</code> .
totaldisk	totaldisk	Returns the total disk space on a system (in Mbytes) and sets <code>SI_TOTALDISK</code> . The total disk space includes all of the operational disks that are attached to a system.

Solaris Live Upgrade (Topics)

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

Chapter 30	Provides overview information on the Solaris Live Upgrade process.
Chapter 31	Provides information that you need to know before creating a boot environment.
Chapter 32	Provides step-by-step instructions for installing Solaris Live Upgrade, using the menus, and creating a boot environment.
Chapter 33	Provides step-by-step instructions for an operating system upgrade or installing a Web Start Flash archive on a boot environment, switching a boot environment to make it active, and recovering quickly from a failed upgrade.
Chapter 34	Provides step-by-step instructions for maintaining a boot environment and viewing status.
Chapter 35	Provides examples of a Solaris Live Upgrade.
Chapter 36	Lists Solaris Live Upgrade commands.

Solaris Live Upgrade (Overview)

This chapter describes the Solaris Live Upgrade process.

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

Solaris Live Upgrade Introduction

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

Solaris Live Upgrade enables you to create copies of a boot environment without affecting the currently running system to do the following:

- Upgrade a system
- Change the disk configuration on the new boot environment to different file system types, sizes, and layouts
- Maintain numerous boot environments with different images. For example, you can create a boot environment that contains current patches and create a boot environment that contains an Update release.

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

Solaris Live Upgrade Process

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

Creating a Boot Environment Overview

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 Systems Overview

Solaris Live Upgrade distinguishes between two file system types: critical file systems and shareable file systems. Critical file systems are required by the Solaris operating environment and are separate mount points in the `vfstab` of the active and inactive boot environments. Examples are root (`/`), `/usr`, `/var`, or `/opt`. These file systems are always copied from the source to the inactive boot environment. Critical file systems are sometimes referred to as *non-shareable*. Shareable file systems are user-defined files such as `/export` that contain the same mount point in the `vfstab` in both the active and inactive boot environments. Therefore, updating shared files in the active boot environment also updates data in the inactive boot environment. When you create a new boot environment, shareable file systems are shared by default. But you can specify a destination slice and then the file systems are copied. For more detailed information on shareable file systems, see “Guidelines for Selecting Slices for Shareable File Systems” on page 245.

Swap is a special circumstance of a shareable file system. Like a shareable file system, all swap slices are shared by default. But, if you specify a destination directory for swap, the swap slice is copied. You achieve this goal by using the character user interface or at the command line by using `lucreate(1M)` with the `-m` option. A limitation to splitting and merging swap slices is that the swap slice cannot be in use by any boot environment except the current boot environment or if the `-s` option is used, the source boot environment. The boot environment creation fails if the swap slice is being used by any other boot environment, whether the slice contains a swap, UFS, or any other file system. A swap slice is not required. For procedures on reconfiguring swap, see “To Create a Boot Environment (Character Interface)” Step 9, or “To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)” on page 262.

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

Copying File Systems

Creating a new boot environment entails copying critical file systems to another slice. The disk might need to be prepared before you create the new boot environment. Check the disk to make sure it is formatted properly:

- Identify slices large enough to hold the file systems to be copied.
- Identify file systems that contain directories that you want to share between boot environments rather than copy. If you want a directory to be shared, you need to create a new boot environment with the directory put on its own slice. The directory is then a file system and can be shared with future boot environments. For more information on creating separate file systems for sharing, see “Guidelines for Selecting Slices for Shareable File Systems” on page 245.

The process of creating a new boot environment begins by identifying an unused slice where the critical file systems can be copied. If a slice is not available or a slice does not meet the minimum requirements, you need to format a new slice. For the procedure on formatting a slice from menus, see “To Create a Boot Environment (Character Interface)” Step 6.

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

For procedures on splitting and merging file systems, see the following procedures:

- “To Create a Boot Environment (Character Interface)” Step 7 or Step 8
- “To Create a Boot Environment and Split File Systems (Command-Line Interface)” on page 260
- “To Create a Boot Environment and Merge File Systems (Command-Line Interface)” on page 259

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 file systems to be copied. When the file systems are copied from the active to the

inactive boot environment, the files are directed to the newly defined directories. The active boot environment is not changed in any way. For procedures on creating a new boot environment, see “Creating a New Boot Environment” on page 251.

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

- Figure 30-1 shows the critical file system root (/) that has been copied to another slice on a disk to create a new boot environment. The active boot environment contains root (/) on one slice. The new boot environment is an exact duplicate with root (/) on a new slice. The file systems /swap and /export/home are shared by the active and inactive boot environments.

Creating a Boot Environment – Copying from root (/) to root (/)

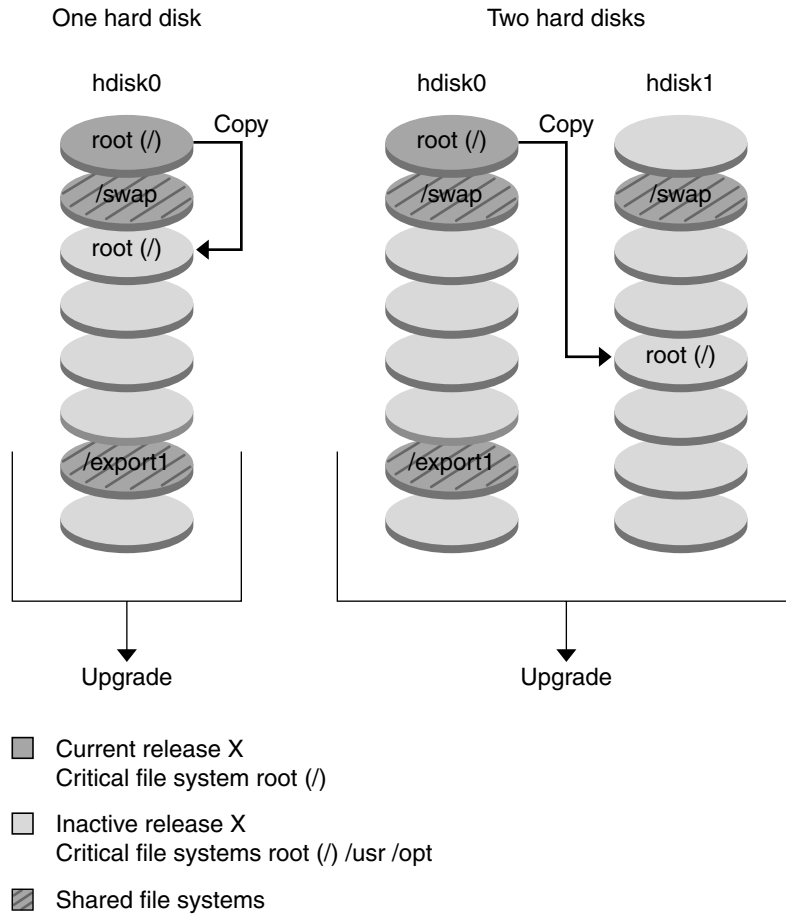


FIGURE 30-1 Creating an Inactive Boot Environment — Copying root (/) to root (/)

- Figure 30-2 shows critical file systems that have been split and been copied to slices on a disk to create a new boot environment. The active boot environment contains root (/) on one slice. On that slice, root (/) contains the /usr, /var, and /opt directories. In the new boot environment, root (/) is split and /usr and /opt are put on separate slices. The file systems /swap and /export/home are shared by both boot environments.

Creating a Boot Environment – Splitting File Systems

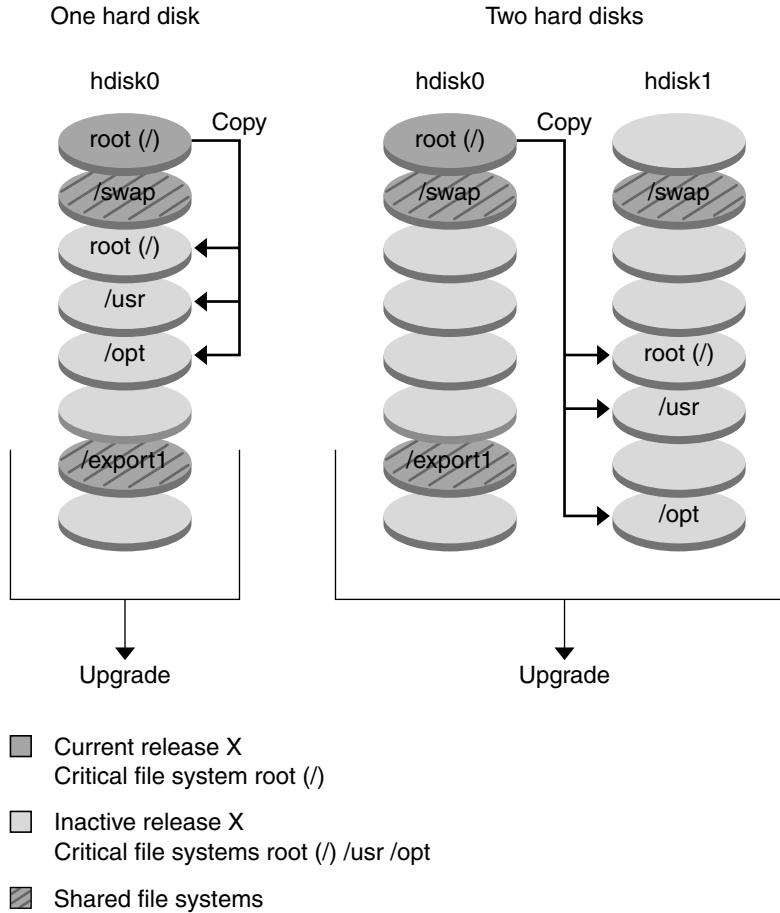


FIGURE 30-2 Creating an Inactive Boot Environment — Splitting File Systems

- Figure 30-3 shows critical file systems that have been merged and been copied to slices on a disk to create a new boot environment. The active boot environment contains root (/), /usr, /var, and /opt each on their own slice. In the new boot environment, /usr and /opt are merged into root (/) on one slice. The file systems /swap and /export/home are shared by both boot environments.

Creating a Boot Environment – Merging File Systems

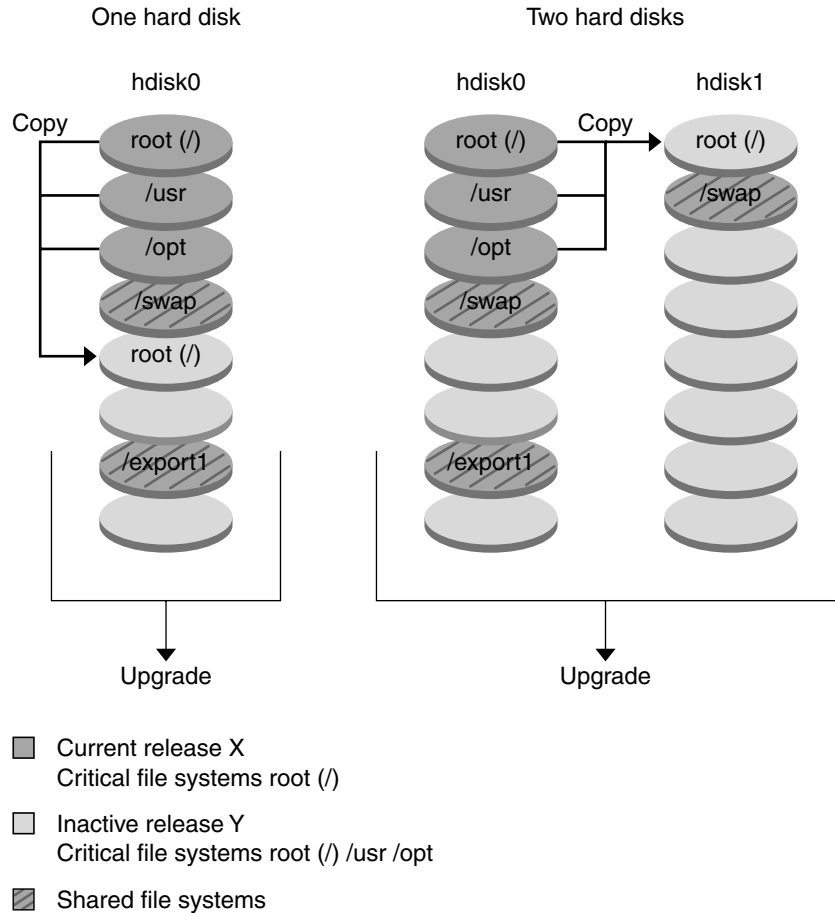


FIGURE 30-3 Creating an Inactive Boot Environment — Merging File Systems

Upgrading a Boot Environment Overview

After you have created a boot environment, it remains unchanged until you are ready to upgrade it. You can perform an upgrade on the boot environment at any time. The upgrade does not affect any files in the active boot environment. When you are ready, you then activate to the new release.

Figure 30-4 shows an upgrade to an inactive boot environment. For procedures on upgrading a boot environment, see Chapter 33.

Upgrading a Boot Environment

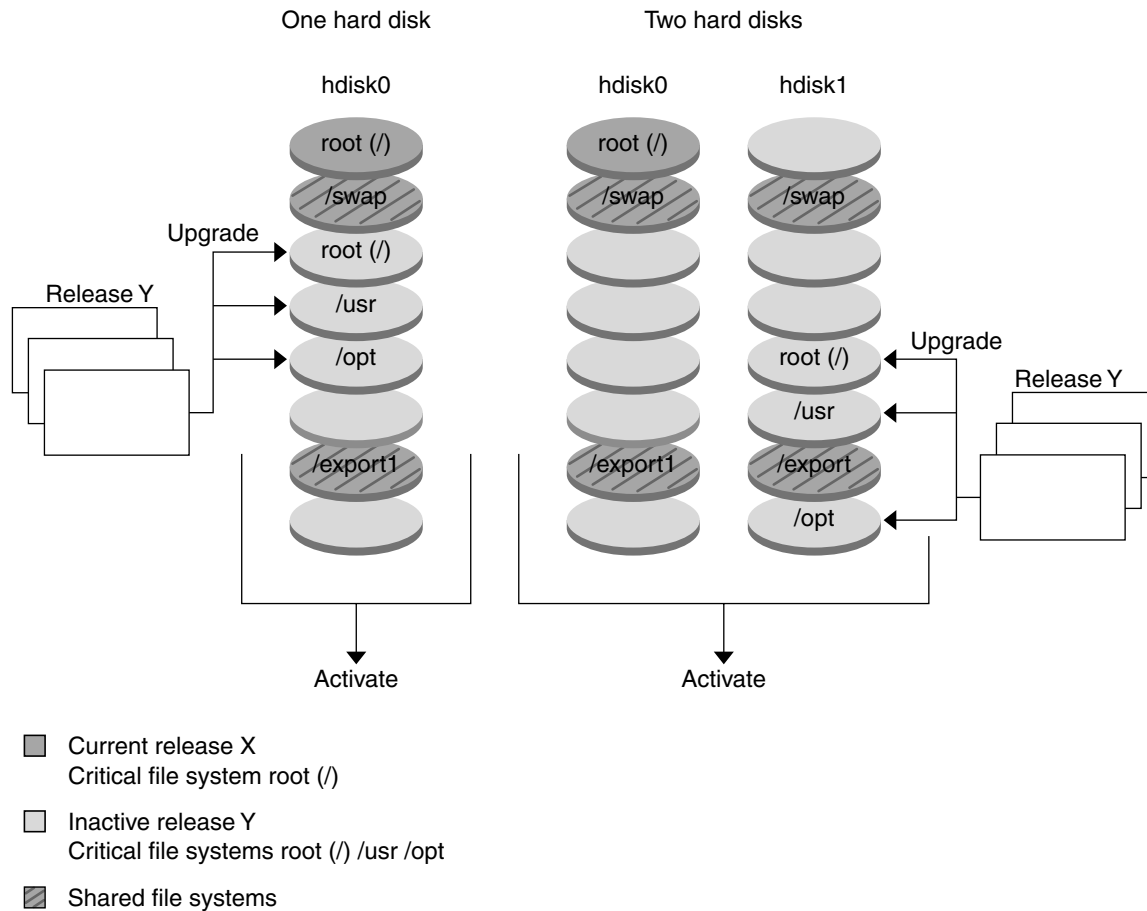


FIGURE 30-4 Upgrading an Inactive Boot Environment

Rather than an upgrade, you can install a Web Start Flash archive on a boot environment. The Web Start Flash installation feature enables you to create a single reference installation of the Solaris operating environment on a system that is called the master system. Then, you can replicate that installation on a number of systems that are called clone systems. In this situation, the inactive boot environment is a clone. For more information about the Web Start Flash installation feature, see Chapter 16.

When you install the Web Start Flash archive on a system, the archive replaces all the files on the existing boot environment as an initial installation would. Figure 30–5 shows an installation of a Web Start Flash archive on an inactive boot environment. For procedures on installing a Web Start Flash archive, see “Installing Web Start Flash Archives on a Boot Environment” on page 278.

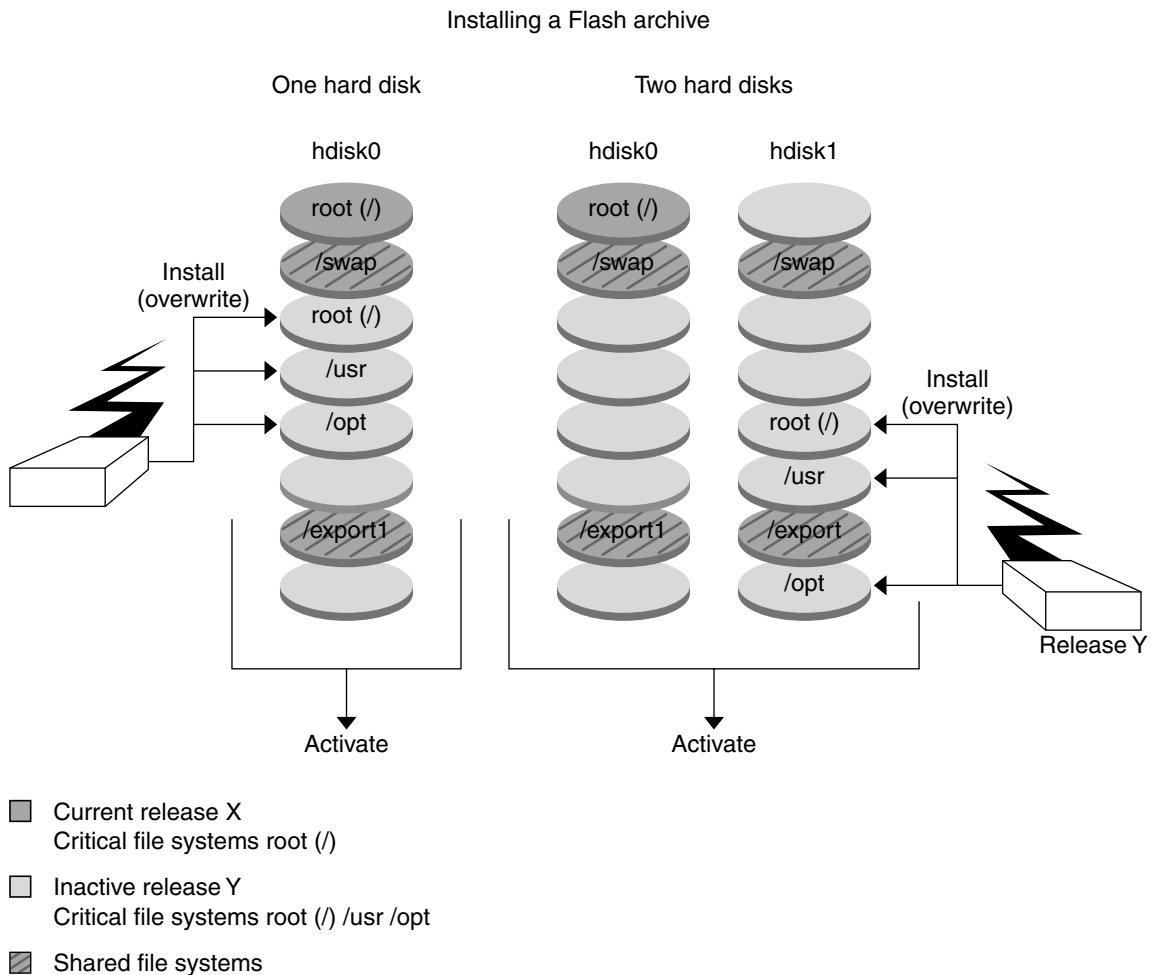


FIGURE 30–5 Installing a Web Start Flash Archive

Activating a Boot Environment Overview

When you are ready to switch and make the new boot environment active, you quickly activate the new boot environment and reboot. When you activate the new boot environment, it is made bootable. Files are synchronized between boot environments the first time that you boot a newly created boot environment.

"Synchronize" here 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. Figure 30-6 shows a switch after a reboot from an inactive to an active boot environment. For procedures on activating a boot environment, see "Activating a Boot Environment" on page 282.

Activating a Boot Environment

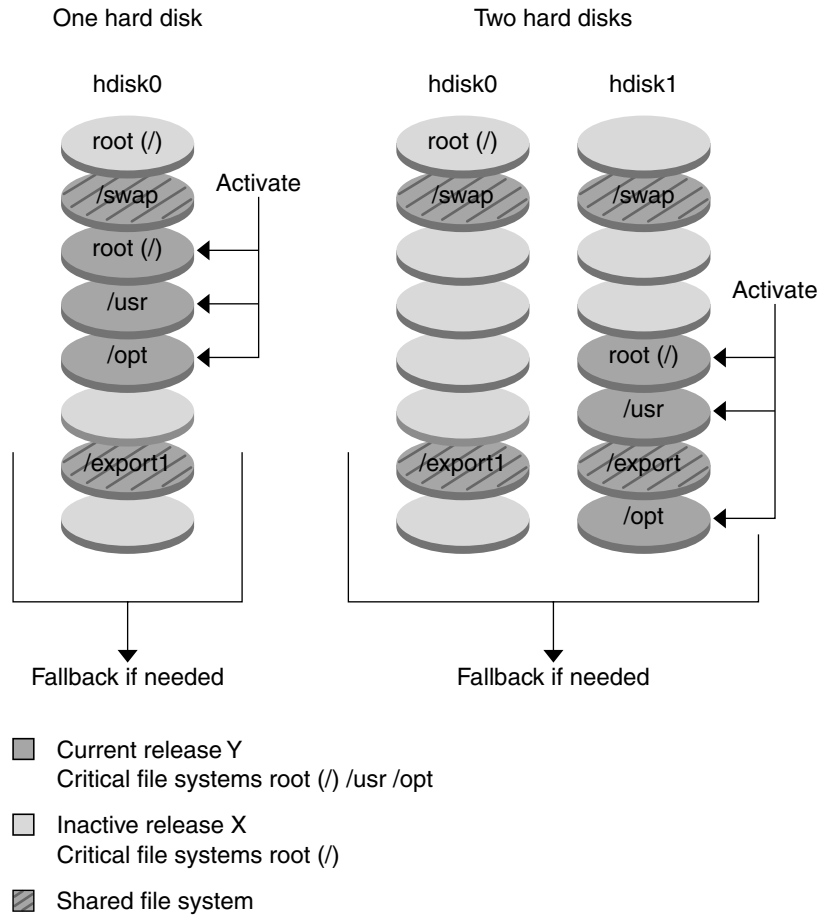


FIGURE 30-6 Activating an Inactive Boot Environment

Fall Back to the Original Boot Environment Overview

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

The use of fallback takes only the time to reboot the system, which is much quicker than backing up and restoring the original. The new boot environment that failed to boot is preserved. The failure can then be analyzed. You can only fall back to the boot environment that was used by `luactivate` to activate the new boot environment.

You fall back to the previous boot environment the following ways:

- If the new boot environment boots successfully, but you are not happy with the results, you run the `luactivate` command with the name of the previous boot environment and reboot.
- If the new boot environment does not boot, you boot the fallback boot environment in single-user mode and run the `luactivate` command and reboot.
- If you cannot boot in single-user mode, you must boot from media or a net installation image, mount the root (/) file system on the fallback boot environment, run the `luactivate` command, and reboot.

For procedures to fall back, see “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 286.

Figure 30-7 shows the switch that is made when you reboot to fallback.

Fallback to Original Boot Environment

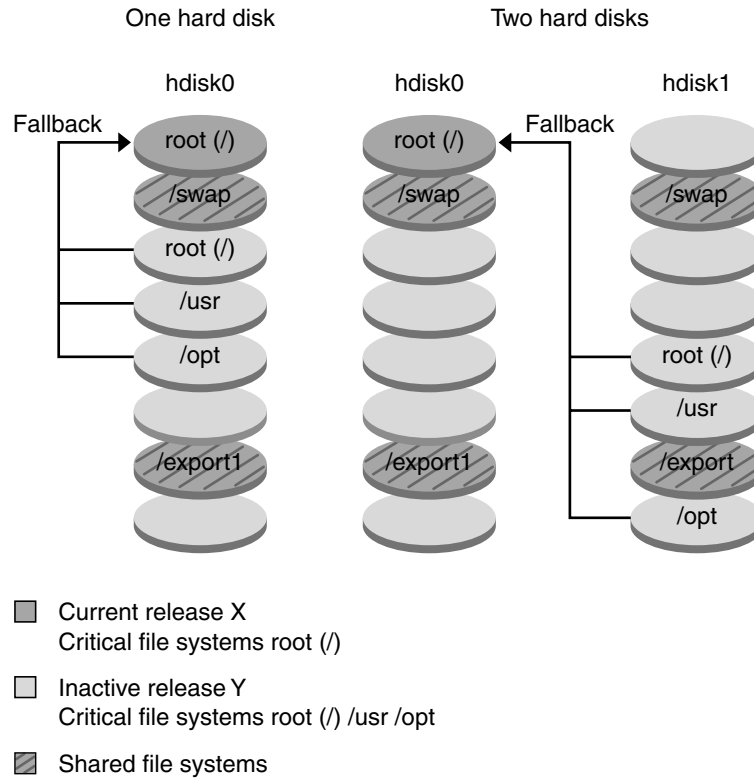


FIGURE 30-7 Fallback to the Original Boot Environment

Maintaining a Boot Environment Overview

You can also do various maintenance activities such as renaming or deleting a boot environment. For maintenance procedures, see Chapter 34.

Solaris Live Upgrade (Planning)

This chapter provides guidelines and requirements for review before installing and using Solaris Live Upgrade. You also should review general information on upgrading in “Checklist for Upgrading” on page 41. This chapter contains the following sections:

- “Solaris Live Upgrade System Requirements” on page 239
- “Solaris Live Upgrade Disk Space Requirements” on page 240
- “Managing Packages and Patches With Solaris Live Upgrade” on page 240
- “Guidelines for Selecting Slices for File Systems” on page 243
- “Using Solaris Live Upgrade From a Remote System” on page 246

Solaris Live Upgrade System Requirements

- You can upgrade from the Solaris 2.6, Solaris 7, or Solaris 8 operating environment to the Solaris 8 or Solaris 9 operating environment.
- You cannot upgrade to the Solaris 7 operating environment.
- You must upgrade to the same release that contained the Solaris Live Upgrade software that you installed on the currently running system. For example, if on your current operating environment, you installed Solaris Live Upgrade from the Solaris 9 release, you must upgrade to the Solaris 9 release.

Solaris Live Upgrade is included in the Solaris 9 software, but if you want to upgrade from previous releases, you need to install the Solaris Live Upgrade packages on your current operating environment. You can install the Solaris Live Upgrade packages from the following:

- An installer on the Solaris 9 DVD, the Solaris 9 Software 2 of 2 CD, or a net installation image.

- The `pkgadd` command. If you are using the `pkgadd` command, the packages are `SUNWlur` and `SUNWluu`, and these packages should be installed in that order.

For instructions on installing the Solaris Live Upgrade software, see “To Install Solaris Live Upgrade” on page 249.

Solaris Live Upgrade Disk Space Requirements

Follow general disk space requirements for an upgrade. See Chapter 5.

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

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

Managing Packages and Patches With Solaris Live Upgrade

The following sections list packages required by Solaris Live Upgrade and provides information on recommended patches. See “Managing Packages and Patches With Solaris Live Upgrade” on page 274 for information on using Solaris Live Upgrade to add packages and patches.



Caution – When upgrading and adding and removing packages or patches, Solaris Live Upgrade requires packages or patches that comply with the SVR4 Advanced Packaging Guidelines. While Sun packages conform to these guidelines, Sun cannot guarantee the conformance of packages from third-party vendors. A nonconformant package can cause the package-addition software during an upgrade to fail or worse, alter the active boot environment.

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

Required Packages

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

TABLE 31-1 Required Packages for Solaris Live Upgrade

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

▼ To Check for Packages on Your System

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

```
% pkginfo [package_name]
```

package_name

List the packages that you want to check

Checking System Patch Levels

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

Guidelines for Creating File Systems With the `lucreate` Command

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

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

Guidelines for Selecting Slices for File Systems

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

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

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

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

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

- Must be a slice from which the system can boot.
- Must meet the recommended minimum size.
- Cannot be a Veritas VxVM volume.
- Can be on different physical disks or the same disk as the active root (/) file system.
- If you have a sun4m system, the root (/) file system cannot be greater than 2 Gbytes.

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

Guidelines for Selecting Slices for root (/) Mirrors and Metadevices

You can use Solaris Live Upgrade on a system that is currently using either Solaris Volume Manager metadevices or Veritas Volume Manager VxVM volumes. The source boot environment can be contained on any combination of physical disk slices, Solaris Volume Manager metadevices, or Veritas Volume Manager volumes. When creating a new boot environment, the slice that is chosen for the root (/) file system for the new boot environment can be either a physical disk slice or a Solaris Volume Manager metadevice. If you choose a Solaris Volume Manager metadevice for the root file system, the metadevice must be either a stripe with only a single disk or a mirror on a single-disk stripe. See `metaroot(1M)` for more details.

You *cannot* use a Veritas VxFS volume for the root (/) file system when creating a new boot environment. For any file system except the root (/) file system, you can use either a physical disk slice, a Solaris Volume Manager metadevice, or a Veritas VxFS volume.

Table 31–2 describes the acceptable disk configurations for creating a boot environment when using metadevices or volumes.

TABLE 31–2 Acceptable Slices for Metadevices or Volumes

Product	Source Slice	If Target Is a Metadevice for a root (/) File System	If Target Is Not a root (/) File System, But Is for /usr, /var, or /opt
Solaris Volume Manger	For the root (/) file system, the source can be a metadevice or physical slice.	The root (/) file system, <i>must</i> be either a stripe with only a single disk or a mirror on a single-disk stripe.	Other file systems can be either a physical slice or metadevice.
Veritas VxVM Volume Manager	For the root (/) file system, the source can be a volume or physical slice.	The root (/) file system, <i>cannot</i> be a VxVM volume; root (/) must be a physical slice.	Other file systems can be either a physical slice or volume.

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

- A physical slice in the form of `/dev/dsk/cnumtnumdnumsnum`
- A Solaris Volume Manager metadevice in the form of `/dev/md/dsk/dnum`
- A Veritas VxFS volume in the form of `/dev/vx/dsk/volume_name`

Using boot environments with metadevices or volumes has special considerations when upgrading or installing a flash archive. See “Upgrading Metadevices and Volumes” on page 273 for details.

Note – If you have problems upgrading with Veritas VxVM, see “System Panics When Upgrading With Solaris Live Upgrade Running Veritas VxVm” on page 344.

Guidelines for Selecting a Slice for a swap File System

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

Guidelines for Selecting Slices for Shareable File Systems

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

- You can reslice the disk before creating the new boot environment and put the shareable file system on its own slice. For example, if root (`/`), `/var`, and `/home` are on the same slice, reconfigure the disk and put `/home` on its own slice. When you create any new boot environments, `/home` is shared with the new boot environment by default.
- If you want to share a directory, the directory must be split off to its own slice. The directory is then a file system that can be shared with another boot environment. You can use the `lucreate` command with the `-m` option to create a new boot environment and split a directory off to its own slice. But, the new file system cannot yet be shared with the original boot environment. You need to run the `lucreate` command with the `-m` option again to create another boot environment. The two new boot environments can then share the directory.

For example, if you wanted to upgrade from the Solaris 8 release to the Solaris 9 release and share `/home`, you could run the `lucreate` command with the `-m` option. You could create a Solaris 8 release with `/home` as a separate file system on its own slice. Then run the `lucreate` command with the `-m` option again to duplicate that boot environment. This third boot environment can then be upgraded to the Solaris 9 release. `/home` is shared between the Solaris 8 and Solaris

9 releases.

For a description of shareable and critical file systems, see “Creating a Boot Environment Overview” on page 226.

Using Solaris Live Upgrade From a Remote System

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

Using Solaris Live Upgrade to Create a Boot Environment (Tasks)

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

- “About Solaris Live Upgrade Interfaces” on page 247
- “Using Solaris Live Upgrade Menus” on page 248
- “Task Map: Installing Solaris Live Upgrade and Creating Boot Environments” on page 249
- “Installing Solaris Live Upgrade” on page 249
- “Starting and Stopping Solaris Live Upgrade (Character Interface)” on page 250
- “Creating a New Boot Environment” on page 251

About Solaris Live Upgrade Interfaces

You can run Solaris Live Upgrade with a character user interface (CUI) or the command-line interface (CLI). The content and sequence of instructions for the CUI and CLI are generally the same. Procedures for both the CUI and CLI are provided in the following sections. The CUI does not provide access to all features of Solaris Live Upgrade. See Chapter 36 for a list of commands and also see the appropriate, associated man pages, which more fully document features.

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

Using Solaris Live Upgrade Menus



FIGURE 32-1 Solaris Live Upgrade Main Menu

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

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

In the following procedures, you might be asked to press a function key. If your function keys do not properly map to the function keys on the Solaris Live Upgrade menus, use Control-F plus the appropriate number.

Task Map: Installing Solaris Live Upgrade and Creating Boot Environments

TABLE 32-1 Task Map: Using Solaris Live Upgrade

Task	Description	For Instructions
Install Solaris Live Upgrade packages	Install packages on your Solaris 2.6, Solaris 7, or Solaris 8 operating environment	“Installing Solaris Live Upgrade” on page 249
Start Solaris Live Upgrade	Start the Solaris Live Upgrade main menu	“Starting and Stopping Solaris Live Upgrade (Character Interface)” on page 250
Create a boot environment	Copy and reconfigure file systems to an inactive boot environment	“Creating a New Boot Environment” on page 251

Installing Solaris Live Upgrade

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

▼ To Install Solaris Live Upgrade

1. **Insert the Solaris 9 DVD or Solaris 9 Software 2 of 2 CD.**
2. **Run the installer for the media you are using.**
 - If you are using the Solaris 9 DVD, change directories to the installer and run the installer.

```
# cd /cdrom/cdrom0/Solaris_9/Tool/Installers
# ./liveupgrade20
```

The Solaris Web Start installer is displayed.
 - If you are using the Solaris 9 Software 2 of 2 CD, run the installer.

```
% ./installer
```

The Solaris Web Start installer is displayed.

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

Starting and Stopping Solaris Live Upgrade (Character Interface)

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

▼ To Start Solaris Live Upgrade Menus

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

1. **Log in as superuser.**
2. **Type:**

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

The Solaris Live Upgrade main menu is displayed.

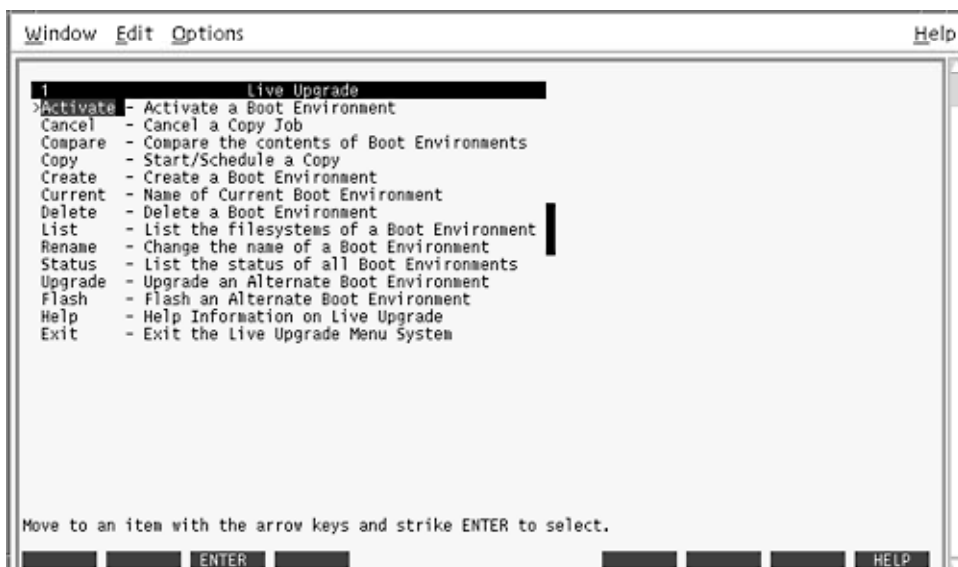


FIGURE 32-2 Solaris Live Upgrade Main Menu

▼ To Stop Solaris Live Upgrade Menus

- From the main menu, select Exit.

Creating a New Boot Environment

You can use Solaris Live Upgrade with menus or the command-line interface (CLI). Procedures are documented for both interfaces. These procedures do not exhaust the possibilities for using Solaris Live Upgrade. For more information about commands, see Chapter 36 and the appropriate man pages, which more fully document the CLI.

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

Before file systems are copied to the new boot environment, they can be customized so that critical file system directories are either merged into their parent directory or split from their parent directory. User-defined (shareable) file systems are shared between

boot environments by default. But shareable file systems can be copied if needed. Swap, which is a shareable file system, can be split and merged also. For an overview of critical and shareable file systems, see “Creating a Boot Environment Overview” on page 226.

This section contains the following procedures:

- “To Create a Boot Environment (Character Interface)” on page 252
- “To Create a Boot Environment for the First Time (Command-Line Interface)” on page 257
- “To Create a Boot Environment and Merge File Systems (Command-Line Interface)” on page 259
- “To Create a Boot Environment and Split File Systems (Command-Line Interface)” on page 260
- “To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)” on page 262
- “To Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)” on page 264
- “To Create a Boot Environment and Copy a Shareable File System (Command-Line Interface)” on page 266
- “To Create a Boot Environment From a Different Source (Command-Line Interface)” on page 267
- “To Create a Boot Environment for a Flash Archive (Command-Line Interface)” on page 268

▼ To Create a Boot Environment (Character Interface)

1. **From the main menu, select Create.**

The system displays the Create a Boot Environment submenu.

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

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

```
Name of Current Boot Environment:  solaris8
Name of New Boot Environment:     solaris9
```

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

The configuration menu appears.

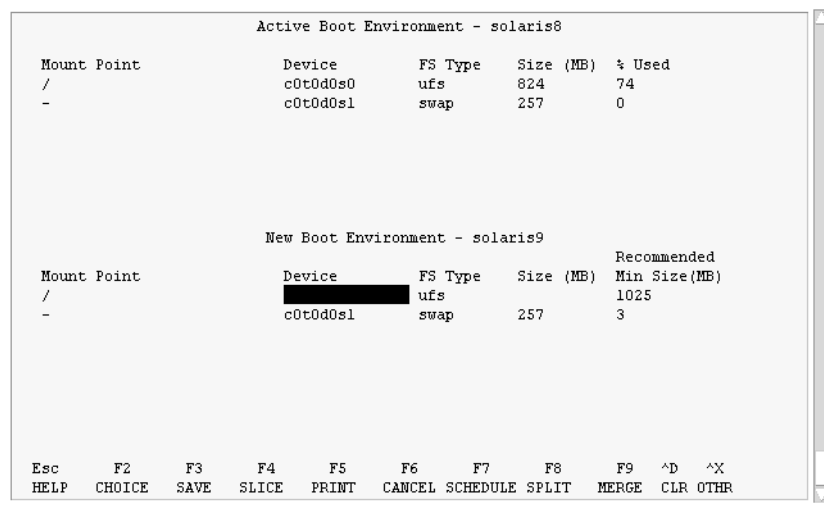


FIGURE 32-3 Solaris Live Upgrade Configuration Menu

The Configuration menu contains the following parts:

- The original boot environment is located at the top of the screen. The boot environment to be created is at the bottom.
- The Device field contains the following information:
 - The name of a disk device is in the form `/dev/dsk/cnumdnumtnumsnnum`.
 - 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 area for selecting a critical file system is blank until you select a critical file system. The critical file systems such as `/usr`, `/var`, or `/opt` can be split or merged with the root (`/`) file system.
 - Shareable file systems such as `/export` or `swap` are displayed in the Device field. These file systems contain the same mount point in both the source and target boot environments. Swap is shared by default, but you can also split and merge (add and remove) swap slices.
For an overview of critical and shareable file systems, see “Creating a Boot Environment Overview” on page 226.
- The FS_Type field enables you to change file system type. The file system type can be one of the following:
 - `vxfs`, which indicates a Veritas file system
 - `swap`, which indicates a swap file system

- `ufs`, which indicates a UFS file system

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

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

5. Select an available slice by pressing F2.

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

a. Use the arrow keys to place the cursor in a field to select a slice or file system type.

- When you place your cursor in the Slice field, all free slices are displayed. For root (/), Choices only displays free slices that meet the root (/) file system limitations. See “Guidelines for Selecting a Slice for the root (/) File System” on page 243.
- When you place your cursor in the FS_Type field, all available file system types are displayed.
- Slices in bold can be selected for the current file system. The size of the slice is estimated by adding the size of the file system plus 30 percent to accommodate an upgrade.
- Slices not in bold are too small to support the given file system. To reslice a disk, see Step 6.

b. Press Return to choose a slice.

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

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

The Solaris Live Upgrade Slice Configuration menu appears.

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

To navigate through this menu, use the arrow keys to move between the Device field and FS_Type field. The Size (Mbytes) field is automatically completed as the devices are selected.

- a. **To free a device, press Control-D.**

The slice is now available and appears on the Choices menu.

- b. **To return to the Configuration menu, press F3.**

7. (Optional) Splitting critical file systems puts the file systems on separate mount points. To split a file system, do the following:

(To merge file systems, see Step 8).

- a. **Select the file system to split.**

You can split or exclude file systems such as `/usr`, `/var`, or `/opt` from their parent directory.

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

- b. **Press F8.**

- c. **Type the file system name for the new boot environment, for example:**

Enter the directory that will be a separate file system on the new boot environment: `/opt`

When the new file system is verified, a new line is added to the screen.

- d. **To return to the Configuration menu, press F3.**

The Configuration menu is displayed.

8. (Optional) Merging puts the file systems on the same mount point. To merge a file system into its parent directory:

(To split file systems, see Step 7.)

- a. **Select the file system to merge.**

You can merge file systems such as `/usr`, `/var`, or `/opt` into their parent directory.

- b. **Press F9.**

The file systems that will be combined are displayed, for example:

`/opt` will be merged into `/`.

- c. **Press Return.**

- d. **To return to the Configuration menu, press F3.**

The Configuration menu is displayed.

9. (Optional) Decide if you want to add or remove swap slices.

- If you want to split a swap slice and put swap on a new slice, continue with Step 10.
- If you want to remove a swap slice, continue with Step 11.

10. (Optional) To split a swap slice, do the following:

- a. In the Device field, select the swap slice that you want to split.
- b. Press F8.
- c. At the prompt, type:

Enter the directory that will be a separate filesystem on the new BE: **swap**

d. Press F2 Choice.

The Choice menu lists the available slices for swap.

e. Select the slice to put swap on.

The slice appears in the Device field and you have a new slice for swap.

11. (Optional) To remove a swap slice, do the following:

- a. In the Device field, select the swap slice that you are removing.
- b. Press F9.
- c. At the prompt, type **y**.

Slice /dev/dsk/c0t4d0s0 will not be swap partition.

Please confirm? [y, n]: **y**

The swap slice no longer exists.

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

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

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


You are notified of the completion by email.

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

You can schedule only one job at a time.

After the creation is complete, the inactive boot environment is ready to be upgraded. See Chapter 33.

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

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

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

1. Log in as superuser.

2. To create the new boot environment, type:

```
# lucreate [-A 'BE_description'] -c BE_name \  
-m mountpoint:device:fs_type [-m mountpoint:device:fs_type] -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.

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

EXAMPLE 32-1 Creating a Boot Environment (Command Line)

In this example, the active boot environment is named `first_disk`. The mount points for the file systems are noted by using the `-m` option. Two file systems are created, root (`/`) and `/usr`. The new boot environment is named `second_disk`. A description, Solaris 9 test Jan. 2001, 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 'Solaris 9 test Jan 2001' -c first_disk
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \
-n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 33.

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

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

1. Log in as superuser.

2. Type:

```
# lucreate [-A 'BE_description'] -m mountpoint:device:fs_type \  
[-m mountpoint:device:fs_type] -m mountpoint:merged:fs_type -n BE_name
```

<code>-A BE_description</code>	(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (<code>BE_name</code>). The description can be any length and can contain any characters.
<code>-m mountpoint:device:fs_type [-m...]</code>	Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed. <ul style="list-style-type: none">■ <code>mountpoint</code> can be any valid mount point or <code>-</code> (hyphen), indicating a swap partition.■ <code>device</code> field can be one of the following:<ul style="list-style-type: none">■ The name of a disk device, of the form <code>/dev/dsk/cnumdnumtnumsnun</code>■ The name of a Solaris Volume Manager metadvice, of the form <code>/dev/md/dsk/dnum</code>■ The name of a Veritas Volume Manager volume, of the form <code>/dev/vx/dsk/volume_name</code>■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent■ <code>fs_type</code> field can be one of the following:<ul style="list-style-type: none">■ <code>ufs</code>, which indicates a UFS file system.■ <code>vxfs</code>, which indicates a Veritas file system.■ <code>swap</code>, which indicates a swap file system. The swap mount point must be a <code>-</code> (hyphen).
<code>-n BE_name</code>	The name of the boot environment to be created. <code>BE_name</code> must be unique on the system.

EXAMPLE 32-2 Creating a Boot Environment and Merge File Systems (Command-Line Interface)

In this example, the file systems on the current boot environment are root (/), /usr, and /opt. The /opt file system is combined with its parent file system /usr. The new boot environment is named second_disk. A description, Solaris 9 test Jan. 2001, is associated with the name second_disk.

```
# lucreate -A 'Solaris 9 test Jan 2001' -c first_disk \  
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \  
-m /usr/opt:merged:ufs -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 33.

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

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

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

1. Log in as superuser.

2. Type:

```
# lucreate [-A 'BE_description'] -m mountpoint:device:fs_type \  
[-m mountpoint:device:fs_type] -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.

<p><code>-m <i>mountpoint:device:</i> <i>fs_type</i> [-m...]</code></p>	<p>Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.</p> <ul style="list-style-type: none"> ■ <i>mountpoint</i> can be any valid mount point or <code>-</code> (hyphen), indicating a swap partition. ■ <i>device</i> field can be one of the following: <ul style="list-style-type: none"> ■ The name of a disk device, of the form <code>/dev/dsk/cnumdnumtnumsnum</code> ■ The name of a Solaris Volume Manager metadvice, of the form <code>/dev/md/dsk/dnum</code> ■ The name of a Veritas Volume Manager volume, of the form <code>/dev/vx/dsk/volume_name</code> ■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent ■ <i>fs_type</i> field can be one of the following: <ul style="list-style-type: none"> ■ <code>ufs</code>, which indicates a UFS file system. ■ <code>vxfs</code>, which indicates a Veritas file system. ■ <code>swap</code>, which indicates a swap file system. The swap mount point must be a <code>-</code> (hyphen).
<p><code>-n <i>BE_name</i></code></p>	<p>The name of the boot environment to be created. <i>BE_name</i> must be unique on the system.</p>

EXAMPLE 32-3 Creating a Boot Environment and Splitting File Systems (Command-Line Interface)

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

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

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

A description, Solaris 9 test Jan. 2001, is associated with the boot environment name `second_disk`.

```
# lucreate -A 'Solaris 9 test Jan 2001' -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
```

EXAMPLE 32-3 Creating a Boot Environment and Splitting File Systems (Command-Line Interface) (Continued)

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 33.

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

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

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

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

1. **Log in as superuser.**

2. **Type:**

```
# lucreate [-A 'BE_description'] -m mountpoint:device:fs_type \  
-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.

<p><code>-m mountpoint:device:</code> <code>fs_type [-m...]</code></p>	<p>Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed.</p> <ul style="list-style-type: none"> ■ <code>mountpoint</code> can be any valid mount point or <code>-</code> (hyphen), indicating a swap partition. ■ <code>device</code> field can be one of the following: <ul style="list-style-type: none"> ■ The name of a disk device, of the form <code>/dev/dsk/cnumdnumtnumsnum</code> ■ The name of a Solaris Volume Manager metadvice, of the form <code>/dev/md/dsk/dnum</code> ■ The name of a Veritas Volume Manager volume, of the form <code>/dev/vx/dsk/volume_name</code> ■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent ■ <code>fs_type</code> field can be one of the following: <ul style="list-style-type: none"> ■ <code>ufs</code>, which indicates a UFS file system. ■ <code>vxfs</code>, which indicates a Veritas file system. ■ <code>swap</code>, which indicates a swap file system. The swap mount point must be a <code>-</code> (hyphen).
<p><code>-n BE_name</code></p>	<p>The name of the boot environment to be created. <code>BE_name</code> must be unique.</p>

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

EXAMPLE 32-4 Creating a Boot Environment and Reconfigure Swap (Command-Line Interface)

In this example, the current boot environment contains root (/) on `/dev/dsk/c0t0d0s0` and swap is on `/dev/dsk/c0t0d0s1`. The new boot environment copies root (/) to `/dev/dsk/c0t4d0s0` and uses both `/dev/dsk/c0t0d0s1` and `/dev/dsk/c0t4d0s1` as swap slices. A description, Solaris 9 test Jan. 2001, is associated with the boot environment name `second_disk`.

```
# lucreate -A 'Solaris 9 test Jan 2001' -c first_disk \
-m /dev/dsk/c0t4d0s0:ufs -m -:/dev/dsk/c0t0d0s1:swap \
-m -:/dev/dsk/c0t4d0s1:swap -n second_disk
```

These swap assignments are effective only after booting from `second_disk`. If you have a long list of swap slices, use the `-M` option. See "To Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)" on page 264.

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 33.

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

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

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

1. **Create a list 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:fs_type \  
-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.

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

EXAMPLE 32-5 Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)

In this example, swap in the new boot environment is the list of slices that are noted in the `/etc/lu/swapslices` file. A description, Solaris 9 test Jan. 2001, is associated with the name `second_disk`.

```
# lucreate -A 'Solaris 9 test Jan 2001' -c first_disk \
-m /:/dev/dsk/c02t4d0s0:ufs -m /usr:/dev/dsk/c02t4d0s1:ufs \
-M /etc/lu/swapslices -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and can be activated (made bootable). See Chapter 33.

▼ To Create a Boot Environment and Copy a Shareable File System (Command-Line Interface)

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

1. Log in as superuser.

2. Create the boot environment.

```
# lucreate [-A 'BE_description'] -m mountpoint:device:fs_type \  
[-m ]mountpoint:device:fs_type -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:fs_type [-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/cnumdnumtnumsnum`
 - 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_type` field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap file system. The swap mount point must be a `-` (hyphen).

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

EXAMPLE 32-6 Creating a Boot Environment and Copying a Shareable File System (Command-Line Interface)

In this example, the current boot environment contains two file systems, root (/) and /home. In the new boot environment, root (/) is split into two file systems, root (/) and /usr. The /home file system is copied to the new boot environment. A description, Solaris 9 test Jan. 2001, is associated with the boot environment name `second_disk`.

```
# lucreate -A 'Solaris 9 test Jan 2001' -c first_disk \  
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \  
-m /home:/dev/dsk/c0t4d0s4:ufs -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 33.

▼ To Create a Boot Environment From a Different Source (Command-Line Interface)

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

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

1. Log in as superuser.

2. Create the boot environment.

```
# lucreate [-A 'BE_description'] -s source_BE_name  
-m mountpoint:device:fs_type -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:fs_type [-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/cnumdnumtnumsnum`
 - 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_type` field can be one of the following:
 - `ufs`, which indicates a UFS file system.
 - `vxfs`, which indicates a Veritas file system.
 - `swap`, which indicates a swap file system. The swap mount point must be a `-` (hyphen).
- `-n BE_name` The name of the boot environment to be created. `BE_name` must be unique on the system.

EXAMPLE 32-7 Creating a Boot Environment From a Different Source (Command-Line Interface)

In this example, a boot environment is created based on the root (`/`) file system in the source boot environment named `third_disk`. `Third_disk` is not the active boot environment. A description, Solaris 9 test Jan. 2001, is associated with the new boot environment named `second_disk`.

```
# lucreate -A 'Solaris 9 test Jan 2001' -s third_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -n second_disk
```

When creation of the new boot environment is complete, it can be upgraded and activated (made bootable). See Chapter 33.

▼ To Create a Boot Environment for a Flash Archive (Command-Line Interface)

The `lucreate` command creates a boot environment that is based on the file systems in the active boot environment. The `lucreate` command with the `-s` option creates an empty boot environment very quickly that could be used to install a Flash archive.

1. Log in as superuser.

2. Create the empty boot environment.

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

<code>-A 'BE_description'</code>	(Optional) Enables the creation of a boot environment description that is associated with the boot environment name (<code>BE_name</code>). The description can be any length and can contain any characters.
<code>-s -</code>	Specifies that an empty boot environment be created.
<code>-m mountpoint:device:fs_type [-m...]</code>	Specifies the file systems' configuration of the new boot environment. The file systems that are specified as arguments to <code>-m</code> can be on the same disk or they can be spread across multiple disks. Use this option as many times as needed to create the number of file systems that are needed. <ul style="list-style-type: none">■ <code>mountpoint</code> can be any valid mount point or <code>-</code> (hyphen), indicating a swap partition.■ <code>device</code> field can be one of the following:<ul style="list-style-type: none">■ The name of a disk device, of the form <code>/dev/dsk/cnumdnumtnumsnum</code>■ The name of a Solaris Volume Manager metadvice, of the form <code>/dev/md/dsk/dnum</code>■ The name of a Veritas Volume Manager volume, of the form <code>/dev/vx/dsk/volume_name</code>■ The keyword <code>merged</code>, indicating that the file system at the specified mount point is to be merged with its parent■ <code>fs_type</code> field can be one of the following:<ul style="list-style-type: none">■ <code>ufs</code>, which indicates a UFS file system.■ <code>vxfs</code>, which indicates a Veritas file system.■ <code>swap</code>, which indicates a swap file system. The swap mount point must be a <code>-</code> (hyphen).
<code>-n BE_name</code>	The name of the boot environment to be created. <code>BE_name</code> must be unique on the system.

EXAMPLE 32-8 Creating a Boot Environment for a Flash Archive (Command-Line Interface)

In this example, a boot environment is created but contains no file systems. A description, Solaris 9 test Jan. 2001, is associated with the new boot environment that is named `second_disk`.

```
# lucreate -A 'Solaris 9 test Jan 2001' -s - \  
-m /:/dev/dsk/c0t4d0s0: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 33.

Upgrading With Solaris Live Upgrade (Tasks)

This chapter explains how to use Solaris Live Upgrade to upgrade and activate an inactive boot environment. It also explains how to recover from an activation failure. This chapter contains the following sections:

- “Task Map: Upgrading a Boot Environment” on page 272
- “Upgrading a Boot Environment” on page 272
- “Installing Web Start Flash Archives on a Boot Environment” on page 278
- “Activating a Boot Environment” on page 282
- “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 286

You can use Solaris Live Upgrade with menus or by using the command-line interface (CLI). Procedures are documented for both interfaces. These procedures do not exhaust the possibilities for using Solaris Live Upgrade. For more information about commands, see Chapter 36 for commands and the appropriate man pages, which more fully document the CLI.

Task Map: Upgrading a Boot Environment

TABLE 33-1 Task Map: Upgrading With Solaris Live Upgrade

Task	Description	For Instructions
Either upgrade a boot environment or install a Web Start Flash archive	<ul style="list-style-type: none">■ Upgrade the inactive boot environment with an operating environment image■ Install a Web Start Flash archive on an inactive boot environment	<ul style="list-style-type: none">■ “Upgrading a Boot Environment” on page 272■ “Installing Web Start Flash Archives on a Boot Environment” on page 278
Activate an inactive boot environment	Makes changes effective and switches the inactive boot environment to active	“Activating a Boot Environment” on page 282
(optional) Switch back if a failure occurs when activating	Reactivates to the original boot environment if a failure occurs	“Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 286

Upgrading a Boot Environment

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

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

Note – If the installation requires more than one CD, you must use the command-line interface procedure for upgrading. See “To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)” on page 276.

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

Rather than upgrading, if you have created a Web Start Flash archive, you could install the archive on an inactive boot environment. The new files overwrite the inactive boot environment critical file systems, but shareable file systems are not changed. See “Installing Web Start Flash Archives on a Boot Environment” on page 278.

Upgrading Metadevices and Volumes

To upgrade or install a flash archive on a new boot environment, the device must be a physical slice. If you have a boot environment that has a file system that is mounted either on a Solaris Volume Manager metadevice or a Veritas file system (VxFS) volume, the upgrade or installation of a flash archive fails. To upgrade or install a flash archive on such a boot environment, you must manually change the boot environment so that all slices are physical disk slices. You could use the `lucreate` command to create another boot environment or you could break the metadevices or volumes.

To use the `lucreate` command to make a copy of the boot environment, you would place the boot environment copy on physical disk slices. For example, suppose your current boot environment’s disk configuration contains the following volumes and slices.

root (/)	Mounted on /dev/md/dsk/d10
/usr	Mounted on /dev/md/dsk/d20
/var	Mounted on /dev/dsk/c0t3d0s0
Free slice	c0t4d0s0
Free slice	c0t4d0s3
Free slice	c0t4d0s4

You could then use the following `lucreate` command to copy the boot environment on physical disk slices. The boot environment could then be upgraded or be installed with a flash archive. In this example, the current boot environment is named, `currentBE`, and the new boot environment is named, `nextBE`.

```
# lucreate -s currentBE -n nextBE -m /:/dev/dsk/c0t4d0s0:ufs \  
-m /usr:/dev/dsk/c0t4d0s3:ufs -m /var:/dev/dsk/c0t4d0s4:ufs
```

You would then be able to upgrade or install a flash archive on the new boot environment, activate it, and then re-mirror or encapsulate it manually.

If you choose to manually break the metadevice or volume, you would need to do the following:

- Manually mount the boot environment
- Edit the `/etc/vfstab` file to change all metadevices or volumes to physical disk slices
- Use the Solaris Volume Manager or Veritas software to tear down the metadevices or volumes
- Unmount the boot environment

The boot environment could then be upgraded or be installed with a flash archive.

Managing Packages and Patches With Solaris Live Upgrade

You can use Solaris Live Upgrade to add patches and packages to a system. As with upgrading a machine, by using Live Upgrade to add patches to a machine, the only downtime the system incurs is that of a reboot. You can add patches and packages directly to a boot environment with the `luupgrade` command. Or, you can apply patches and packages to a master system and apply a Web Start Flash archive of that system to a boot environment by using Live Upgrade.

To add patches to a boot environment directly, create a new boot environment and use the `luupgrade` command with the `-t` option. To add packages to a boot environment, use the `luupgrade` command with the `-p` option. For more information, see the man page, `luupgrade(1M)`.

Or, you can use Solaris Live Upgrade to install a Web Start Flash archive that contains a complete copy of a boot environment with new packages and patches already included. This complete boot environment or single reference system is called a master system. The process of creating a Web Start Flash archive begins with creating a master system. After you have created a master system, add any patches and packages that you want to install. Then, create a Web Start Flash archive of the master system. Use Solaris Live Upgrade to install the archive on the new boot environment. You can copy the boot environment, change it, and distribute it as many times as necessary. For details about how to create a Web Start Flash archive, see Chapter 18. For information on using Solaris Live Upgrade to install a Web Start Flash archive, see “Installing Web Start Flash Archives on a Boot Environment” on page 278.



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

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

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

To upgrade by using this procedure, you must use only one CD, a DVD, or a combined installation image. For an installation that requires more than one CD, you must use the procedure: “To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)” on page 276.

1. **From the Solaris Live Upgrade main menu, select Upgrade.**
The Upgrade menu screen is displayed.
2. **Type the new boot environment’s name.**
3. **Type the path to where the Solaris installation image is located.**

Installation Media Type	Description
Network File System	Specify the path to the network file system where the installation image is located.
Local file	Specify the path to the local file system where the installation image is located.
Local tape	Specify the local tape device and the position on the tape where the installation image is located.
Local device, DVD, or CD	Specify the local device and the path to the installation image.

- If you are using DVD or one CD, type the path to that disc, as in this example:
`Package Media: /cdrom/solaris9/s0`
- If you have a combined image on the network, type the path to the network file system as in this example:
`Package Media: /net/installmachine/export/solaris9/os_image`

4. To upgrade, press F3.
5. When the upgrade is completed, the main menu is displayed.

▼ To Upgrade an Operating System Image on a Boot Environment (Command-Line Interface)

To upgrade by using this procedure, you must use only one CD, a DVD, or a combined installation image. If the installation requires more than one CD, you must use the procedure: “To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface)” on page 276.

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

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

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

EXAMPLE 33-1 Upgrading an Operating System Image on a Boot Environment (Command-Line Interface)

In this example, the `second_disk` boot environment is upgraded.

```
# luupgrade -u -n second_disk \  
-s /net/installmachine/export/solaris9/OS_image
```

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

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

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

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

<code>-u</code>	Upgrades an operating system image on a boot environment
<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

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

4. This step is identical to the previous step, but the `-u` option is replaced by the `-i` option. Also, choose to run the installer on the second CD with menus or with text.

- This command runs the installer on the second CD with menus.

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

- This command runs the installer on the second CD with text and requires no user interaction.

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

<code>-i</code>	Looks for an installation program on the specified medium and runs that program. The installer program is specified with <code>-s</code> .
<code>-n <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.
<code>-O '-nodisplay -noconsole'</code>	(Optional) Runs the installer on the second CD in text mode and requires no user interaction.

The boot environment is ready to be activated. See “Activating a Boot Environment” on page 282.

EXAMPLE 33-2 Upgrading an Operating System Image From Multiple CDs (Command-Line Interface)

In this example, the `second_disk` boot environment is upgraded and the installation image is on two CDs: the Solaris 9 Software 1 of 2 and the Solaris 9 Software 2 of 2 CDs. The `-u` option determines if sufficient space for all the packages is on the CD set. The `-O` option with the `-nodisplay` and `-noconsole` options prevents the character user interface from displaying after the reading of the second CD. If you use these options, you are not prompted to input information. Omit these options to display the interface.

Insert the Solaris 9 Software 1 of 2 CD and type:

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

Insert the Solaris 9 Software 2 of 2 CD and type:

EXAMPLE 33-2 Upgrading an Operating System Image From Multiple CDs (Command-Line Interface) *(Continued)*

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

Note – Use the `luupgrade` command with the `-i` option to install any additional CDs.

Installing Web Start Flash Archives on a Boot Environment

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

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

Note – For examples of the correct syntax for paths that are associated with archive storage, see “`archive_location` Keyword” on page 196.

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

To use the Web Start Flash installation feature, you need to have installed the master system and to have created the Web Start Flash archive. For more information about Web Start Flash, see Chapter 16.

Upgrading Metadevices and Volumes

To upgrade or install a flash archive on a new boot environment, the device must be a physical slice. If you have a boot environment that has a file system that is mounted either on a Solaris Volume Manager metadevice or a Veritas file system (VxFS)

volume, the upgrade or installation of a flash archive fails. To upgrade or install a flash archive on such a boot environment, you must manually change the boot environment so that all slices are physical disk slices. You could use the `lucreate` command to create another boot environment or you could tear down the metadevices or volumes.

To use the `lucreate` command to make a copy of the boot environment, you would place the boot environment copy on physical disk slices. For example, suppose your current boot environment's disk configuration is the following:

```
root (/)          Mounted on /dev/md/dsk/d10
/usr             Mounted on /dev/md/dsk/d20
/var            Mounted on /dev/dsk/c0t3d0s0
Free slice      c0t4d0s0
Free slice      c0t4d0s3
Free slice      c0t4d0s4
```

You could then use the following `lucreate` command to copy the boot environment on physical disk slices. The boot environment could then be upgraded or be installed with a flash archive. In this example, the current boot environment is named `currentBE` and the new boot environment is named `nextBE`.

```
# lucreate -s currentBE -n nextBE -m /:/dev/dsk/c0t4d0s0:ufs \
-m /usr:/dev/dsk/c0t4d0s3:ufs -m /var:/dev/dsk/c0t4d0s4:ufs
```

You would then be able to upgrade or install a flash archive on the new boot environment, activate it, and then re-mirror or encapsulate it manually.

If you choose to manually tear down the metadvice or volume, you would need to do the following:

- Manually mount the boot environment
- Edit the `/etc/vfstab` file to change all metadevices or volumes to physical disk slices
- Use the Solaris Volume Manager or Veritas software to tear down the metadevices or volumes
- Unmount the boot environment

The boot environment could then be upgraded or be installed with a flash archive.

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

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

The Flash an Inactive Boot Environment menu is displayed.

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

```
Name of Boot Environment: solaris_9
Package media: /net/install-svr/export/s9/latest
```

3. Press F1 to add an archive.

An Archive Selection submenu displays an empty list, the name of a single archive, or a list of a number of archives. If more than one archive is listed, you have layered archives. For an explanation of layered archives, see “Installing Layered Web Start Flash Archives” on page 112.

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

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

- To install the list with no changes, proceed to Step 4.
- If the Web Start Flash archives list is empty or contains an archive that you do not want to install, proceed to Step a.

- a. You can add or remove archives.

- Press F1 to add an archive to the list. Proceed to Step b.

Note – If you have a list of more than one archive, you have layered archives. For more information, see “Installing Layered Web Start Flash Archives” on page 112.

A Select Retrieval Method submenu is displayed.

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

- Press F2 to remove an archive from the list. Proceed to Step e.

- b. On the Select Retrieval Method menu, select the location of the Web Start Flash archive.

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

A Retrieval submenu is displayed, similar to the following example, which depends on the media you selected.

NFS Location:

- c. **Type the path to the archive, as in the following example.**

NFS Location: `host:/path/to archive.flar`

- d. **Press F3 to add the archive to the list.**

- e. **When the list contains the archives that you want to install, press F6 to exit.**

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

The Web Start Flash archive or layered archives are installed on the boot environment. All files on the boot environment are overwritten, except for shareable files.

The boot environment is ready for activation. See “To Activate a Boot Environment (Character Interface)” on page 283.

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

1. **Log in as superuser.**

2. **Type:**

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

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

-f	Indicates to upgrade 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.
-J ' <i>profile</i> '	Entry from a JumpStart profile that is configured for a flash installation. See <code>pfinstall(1M)</code> and Chapter 21 for information on the JumpStart software. You must specify one of -a, -j, or -J.
-j <i>profile_path</i>	Path to a JumpStart profile that is configured for a flash installation. See <code>pfinstall(1M)</code> and Chapter 21 for information on the JumpStart software. You must specify one of -a, -j, or -J.
-a <i>archive</i>	Path to the flash archive when the archive is available on the local file system. You must specify one of -a, -j, or -J.

EXAMPLE 33-3 Installing Web Start Flash Archives on a Boot Environment (Command-Line Interface)

In this example, a flash archive is installed on the `second_disk` boot environment. The -J option is used to retrieve the archives. All files are overwritten on `second_disk` except shareable files.

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-J 'archive_location http://example.com/myflash.flar'
```

Activating a Boot Environment

Activating a boot environment makes it bootable on the next reboot of the system. You can also switch back quickly to the original boot environment if a failure occurs on booting the newly active boot environment. See “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 286.

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

- The boot environment must have a status of “complete.” To check status, see “Displaying the Status of All Boot Environments” on page 293.
- If the boot environment is not the current boot environment, you cannot have mounted the partitions of that boot environment by using `lumount(1M)` or `mount(1M)`.
- The boot environment that you want to activate cannot be involved in a comparison operation. See “Comparing Boot Environments” on page 297.

Note – If you want to reconfigure swap, make this change prior to booting the inactive boot environment. By default, all boot environments share the same swap devices. To reconfigure swap, see “To Create a New Boot Environment,” Step 9 or “To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)” on page 262.

▼ To Activate a Boot Environment (Character Interface)

The first time you boot from a newly created boot environment, Solaris Live Upgrade software synchronizes the new boot environment with the boot environment that was last active. The active boot environment is not necessarily the boot environment that was the source for the newly created boot environment. "Synchronize" here means that certain system files and directories are copied from the last-active boot environment to the boot environment being booted. Solaris Live Upgrade does not perform this synchronization after this initial boot, unless you type **yes** when asked the question "Do you want to force a Live Upgrade sync operations?"

If you force a synchronization, Solaris Live Upgrade checks for conflicts between files that are subject to synchronization. When the new boot environment is booted and a conflict is detected, a warning is issued and the files are not synchronized. Activation can be completed successfully, in spite of such a conflict. A conflict can occur if you, for example, do one of the following:

- Upgrade the current boot environment to a new operating system
- Upgrade an inactive boot environment to a new operating system and modify files on the original boot environment
- Modify system files such as `/etc/passwd` on both the boot environments

Use this option with great care, because you might not be aware or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 9 software on your current boot environment and booted back to a Solaris 2.6 release with a forced synchronization, files could be changed on the 2.6 release. Because files are dependent on the release of the operating environment, the boot to the Solaris 2.6 release could fail because the Solaris 9 files might not be compatible with the Solaris 2.6 files.

1. From the Solaris Live Upgrade main menu, select Activate.

2. Type the name of the boot environment to make active:

```
Name of Boot Environment: solaris_9
Do you want to force a Live Upgrade sync operations: no
```

3. You can either continue or force a synchronization of files.

- Press Return to continue.

The first time that the boot environment is booted, files are automatically synchronized.

- You can force a synchronization of files, but use this feature with caution. Operating systems on each boot environment must be compatible with files that are being synchronized. To force a synchronization of files, type:

```
Do you want to force a Live Upgrade sync operations: yes
```

4. Press F3 to begin the activation process.

5. Press Return to continue.

The new boot environment is activated at the next reboot.

6. To activate the inactive boot environment, reboot:

```
# init 6
```

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

1. Log in as superuser.

2. To activate the boot environment, type:

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

BE_name Specifies the name of the boot environment that is to be activated

3. Reboot.

```
# init 6
```



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

EXAMPLE 33–4 Activating a Boot Environment (Command-Line Interface)

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

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

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

The first time you boot from a newly created boot environment, Solaris Live Upgrade software synchronizes the new boot environment with the boot environment that was last active. The active boot environment is not necessarily the boot environment that was the source for the newly created boot environment. "Synchronize" here means that certain system files and directories are copied from the last-active boot environment to the boot environment being booted. Solaris Live Upgrade does not perform this synchronization after this initial boot, unless you use the `luactivate` with the `-s` option.

If you force a synchronization with the `-s` option, `luactivate` 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, for example, do the following:

- Upgrade the current boot environment to a new operating system
- Upgrade an inactive boot environment to a new operating system and modify files on the original boot environment
- Modify system files such as `/etc/passwd` on both the boot environments

Use this option with great care, because you might not be aware or in control of changes that might have occurred in the last-active boot environment. For example, if you were running Solaris 9 software on your current boot environment and booted back to a Solaris 2.6 release with a forced synchronization, files could be changed on the 2.6 release. Because files are dependent on the release of the operating environment, the boot to the Solaris 2.6 release could fail because the Solaris 9 files might not be compatible with the Solaris 2.6 files.

1. Log in as superuser.

2. To activate the boot environment, type:

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

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

-s

Causes synchronization of files between the last-active boot environment and the new boot environment.

The -s option is used to synchronize files after the first activation of a boot environment. The first time that a boot environment is activated, the files between the boot environment are synchronized, but with subsequent activations, the files are not synchronized. Use this option with great caution, because you might not be aware or in control of changes that might have occurred in the last-active boot environment.

3. Reboot.

```
# init 6
```

EXAMPLE 33-5 Activating a Boot Environment (Command-Line Interface)

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

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

Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)

If a failure is detected after upgrading or if the application is not compatible with an upgraded component, fall back to the original boot environment by using one of the following procedures, depending on your platform. The first procedure runs `luactivate` to switch back to the original boot environment. If the first procedure fails, use the second procedure to boot from a DVD-ROM, CD-ROM, or net image.

▼ To Fall Back Despite Successful New Boot Environment Creation

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

1. **Log in as superuser.**
2. **Type:**

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

BE_name Specifies the name of the boot environment to be activated

3. Reboot.

```
# init 6
```

The previous working boot environment becomes the active boot environment.

▼ 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 “To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 288.

1. At the OK prompt, boot the machine to single-user state from the Solaris 9 DVD, Solaris 9 Installation CD, the network, or a local disk:

```
OK boot cdrom -s
```

or

```
OK boot net -s
```

or

```
OK boot device_name -s
```

device_name Specifies the name of devices from where the system can boot, for example /dev/dsk/c0t0d0s0

2. Type:

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

BE_name Specifies the name of the boot environment to be activated

- If this command fails to display a prompt, proceed to “To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 288.
- If the prompt is displayed, continue.

3. At the prompt, type:

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

A message displays that the fallback activation is successful.

4. Reboot.

```
# init 6
```

The previous working boot environment becomes the active boot environment.

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

Use this procedure to boot from a DVD, CD, or a net installation image. You need to mount the root (/) slice from the last active boot environment. Then run the `luactivate` command, which makes the switch. When you reboot, the last active boot environment is up and running again.

1. **At the OK prompt, boot the machine to single-user state from the Solaris 9 DVD, Solaris 9 Installation CD, the network, or a local disk:**

```
OK boot cdrom -s
```

or

```
OK boot net -s
```

or

```
OK boot device_name -s
```

device_name Specifies the name of the disk and the slice where a copy of the operating system resides, for example `/dev/dsk/c0t0d0s0`

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

```
# fsck device_name
```

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

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

```
# mount device_name /mnt
```


device_name

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

4. From the active boot environment root (/) slice, type:

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

`luactivate` activates the previous working boot environment and indicates the result.

5. Unmount /mnt

```
# umount /mnt
```

6. Reboot.

```
# init 6
```

The previous working boot environment becomes the active boot environment.

Maintaining Solaris Live Upgrade Boot Environments (Tasks)

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

- “Overview of Solaris Live Upgrade Maintenance” on page 292
- “Displaying the Status of All Boot Environments” on page 293
- “Updating a Previously Configured Boot Environment” on page 294
- “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 296
- “Comparing Boot Environments” on page 297
- “Deleting an Inactive Boot Environment” on page 299
- “Displaying the Name of the Active Boot Environment” on page 300
- “Changing the Name of a Boot Environment” on page 301
- “Adding or Changing a Description Associated With a Boot Environment Name” on page 302
- “Viewing the Configuration of a Boot Environment” on page 305

Overview of Solaris Live Upgrade Maintenance

TABLE 34-1 Overview of Solaris Live Upgrade Maintenance

Task	Description	For Instructions
(Optional) 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 293 ■ “Comparing Boot Environments” on page 297 ■ “Displaying the Name of the Active Boot Environment” on page 300 ■ “Viewing the Configuration of a Boot Environment” on page 305
(Optional) Update an inactive boot environment	Copy file systems from the active boot environment again without changing the configuration of file systems	“Updating a Previously Configured Boot Environment” on page 294
(Optional) Other tasks	<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 ■ Cancel scheduled jobs 	<ul style="list-style-type: none"> ■ “Deleting an Inactive Boot Environment” on page 299 ■ “Changing the Name of a Boot Environment” on page 301 ■ “Adding or Changing a Description Associated With a Boot Environment Name” on page 302 ■ “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 296

Displaying the Status of All Boot Environments

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

The following details for each boot environment are displayed:

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

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

- From the main menu, select Status.

A table that is similar to the following is displayed:

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

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

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

1. Log in as superuser.
2. Type:

```
# lustatus BE_name
```

BE_name Specifies the name of the inactive boot environment to view status. If *BE_name* is omitted, `lustatus` displays status for all boot environments in the system.

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

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

Note – You could not perform copy, rename, or upgrade operations on `disk_b_S8` because it is not complete, nor on `disk_b_S7database` because a live upgrade operation is 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 293 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 296.

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

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

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

3. Continue or schedule the copy to occur later:

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

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

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

The inactive boot environment is updated.

To cancel a scheduled copy, see “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 296.

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

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

1. Log in as superuser.
2. Type:

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

-n BE_name Specifies the name of the boot environment that has file systems that are to be replaced.

<code>-s source_BE</code>	(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, <code>lumake</code> uses the current boot environment as the source.
<code>-t time</code>	(Optional) Set up a batch job to copy over file systems on a specified boot environment at a specified time. The time is given in the format that is specified by the man page, <code>at(1)</code> .
<code>-m email_address</code>	(Optional) Enables you to email <code>lumake</code> output to a specified address on command completion. <code>email_address</code> is not checked. You can use this option only in conjunction with <code>-t</code> .

EXAMPLE 34-1 Updating a Previously Configured Boot Environment (Command-Line Interface)

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

```
# lumake -n second_disk -s first_disk -m joe@anywhere.com
```

The files on `first_disk` are copied to `second_disk` and email is sent for notification. To cancel a scheduled copy, see “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 296.

Canceling a Scheduled Create, Upgrade, or Copy Job

A boot environment’s scheduled creation, upgrade, or copy job can be canceled just prior to the time the job starts. A job can be scheduled for a specific time either in the GUI with the Create a Boot Environment, Upgrade a Boot Environment, or Copy a Boot Environment menus. In the CLI, the job can be scheduled by the `lumake` command. At any time, only one job can be scheduled on a system.

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

1. From the main menu, select **Cancel**.
2. To view a list of boot environments that is available for canceling, press **F2**.
3. Select the boot environment to cancel.

The job no longer executes at the time specified.

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

1. Log in as superuser.

2. Type:

```
# lucancel
```

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

Comparing Boot Environments

Use the Compare menu or `lucompare` to check for differences between the active boot environment and other boot environments. To make a comparison, the inactive boot environment must be in a complete state and cannot have a copy job that is pending. See “Displaying the Status of All Boot Environments” on page 293.

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

▼ To Compare Boot Environments (Character Interface)

1. From the main menu, select Compare.

2. Select either Compare to Original or Compare to an Active Boot Environment.

3. Press F3.

4. Type the names of the original (active) boot environment, the inactive boot environment, and the path to a file:

```
Name of Parent: solaris8
```

```
Name of Child: solaris8-1
```

```
Full Pathname of the file to Store Output: /tmp/compare
```

5. To save to the file, press F3.

The Compare menu displays the following file attributes:

- Mode.
- Number of links.
- Owner.

- Group.
- Checksum – Computes checksums only if the file in the specified boot environment matches its counterpart on the active boot environment in all of the fields that are described previously. If everything matches but the checksums differ, the differing checksums are appended to the entries for the compared files.
- Size.
- Existence of files in only one boot environment.

6. To return to the Compare menu, press F3.

▼ To Compare Boot Environments (Command-Line Interface)

1. Log in as superuser.

2. Type:

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

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

EXAMPLE 34-2 Comparing Boot Environments (Command-Line Interface)

In this example, `first_disk` boot environment (source) is compared to `second_disk` boot environment and the results are sent to a file.

```
# /usr/sbin/lucompare -i /etc/lu/compare/ \
-o /var/tmp/compare.out second_disk
```

Deleting an Inactive Boot Environment

Use either the Delete menu or `ludelete`. You cannot delete the active boot environment or the boot environment that is activated on the next reboot. The boot environment to be deleted must be complete. A complete boot environment is not participating in an operation that will change its status. Use “Displaying the Status of All Boot Environments” on page 293 to determine a boot environment’s status. Also, you cannot delete a boot environment that has file systems mounted with `lumount`.

▼ To Delete an Inactive Boot Environment (Character Interface)

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

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

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

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

1. Log in as superuser.
2. Type:

```
# ludelete BE_name
```

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

EXAMPLE 34-3 Deleting an Inactive Boot Environment (Command-Line Interface)

In this example, the boot environment, `second_disk`, is deleted.

```
# ludelete second_disk
```

Displaying the Name of the Active Boot Environment

Use the Current menu or the `lucurr` command to display the name of the currently running boot environment. If no boot environments are configured on the system, the message "No Boot Environments are defined" is displayed. Note that `lucurr` reports only the name of the current boot environment, not the boot environment that is active on the next reboot. See "Displaying the Status of All Boot Environments" on page 293 to determine a boot environment's status.

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

- From the main menu, select **Current**.

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

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

- Type:

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

EXAMPLE 34-4 Displaying the Name of the Active Boot Environment (Command-Line Interface)

In this example, the name of the current boot environment is displayed.

```
# /usr/sbin/lucurr  
solaris8
```

Changing the Name of a Boot Environment

Renaming a boot environment is often useful when you upgrade the boot environment from one Solaris release to another release. For example, following an operating system upgrade, you might rename the boot environment `solaris7` to `solaris8`. Use the Rename menu or `lurename` command to change the inactive boot environment's name.

The new name must adhere to the following limitations.

- Not exceed 30 characters in length.
- Consist only of alphanumeric characters and other ASCII characters that are not special to the UNIX shell. See the "Quoting" section of `sh(1)`.
- Contain only single-byte, 8-bit characters.
- Be unique on the system.

A boot environment must have the status "complete" before you rename it. See "Displaying the Status of All Boot Environments" on page 293 to determine a boot environment's status. You cannot rename a boot environment that has file systems mounted with `lumount` or `mount`.

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

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

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

1. Log in as superuser.
2. Type:

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

`-e BE_name` Specifies the inactive boot environment name to be changed
`-n new_name` Specifies the new name of the inactive boot environment

In this example, `second_disk` is renamed to `third_disk`.

```
# lurename -e second_disk -n third_disk
```

Adding or Changing a Description Associated With a Boot Environment Name

You can associate a description with a boot environment name. The description never replaces the name. Although a boot environment name is restricted in length and characters, the description can be of any length and of any content. The description can be simple text or as complex as a `gif` file. You can create this description at these times:

- When you create a boot environment with the `lucreate` command and use the `-A` option
- After the boot environment has been created by using the `ludesc` command

For more information on using the `-A` option with `lucreate`, see “To Create a Boot Environment for the First Time (Command-Line Interface)” on page 257. For more information on creating the description after the boot environment has been created, see the following procedures or the man page, `ludesc(1M)`.

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

1. **Log in as superuser.**

2. **Type:**

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

`-n BE_name 'BE_description'` Specifies the boot environment name and the new description to be associated with the name

EXAMPLE 34-5 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 a text file that is enclosed in single quotes.

```
# /usr/sbin/ludesc -n second_disk 'Solaris 9 test build, January 2001.'
```

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

1. Log in as superuser.
2. Type:

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

<code>-n BE_name</code>	Specifies the boot environment name and the new description to be associated with the name
<code>-f file_name</code>	Specifies the file to be associated with a boot environment name

EXAMPLE 34-6 Adding a Description to a Boot Environment Name With a File

In this example, a boot environment description is added to a boot environment that is named `second_disk`. The description is contained in a `gif` file.

```
# /usr/sbin/ludesc -n second_disk -f rose.gif
```

▼ To Determine a Boot Environment Name From a Text Description

1. Log in as superuser.
2. Type:

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

<code>-A 'BE_description'</code>	Displays the boot environment name that is associated with the description
----------------------------------	--

EXAMPLE 34-7 Determining a Boot Environment Name From a Description

In this example, the name of the boot environment, `second_disk`, is determined by using the `-A` option with the description.

EXAMPLE 34-7 Determining a Boot Environment Name From a Description (Continued)

```
# /usr/sbin/ludesc -A 'Solaris 9 test build, January 2001.'
second_disk
```

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

1. Log in as superuser.

2. Type:

```
# /usr/sbin/ludesc -f file_name
```

`-f file_name` Displays the boot environment name that is associated with the description that is located in a file

EXAMPLE 34-8 Determining a Boot Environment Name From a Description in a File

In this example, the name of the boot environment, `second_disk`, is determined by using the `-f` option and the name of the file that contains the description.

```
# /usr/sbin/ludesc -f rose.gif
second_disk
```

▼ To Determine a Boot Environment Description From a Name

1. Log in as superuser.

2. Type:

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

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

EXAMPLE 34-9 Determining a Boot Environment Description From a Name

In this example, the description is determined by using the `-n` option with the boot environment name.

```
# /usr/sbin/ludesc -n second_diskSolaris 9 test build, January 2001
```

Viewing the Configuration of a Boot Environment

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

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

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

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

3. Press F3.

The following example displays a list.

Filesystem	fstype	size(Mb)	Mounted on
/dev/dsk/c0t0d0s1	swap	512.11	-
/dev/dsk/c0t4d0s3	ufs	3738.29	/
/dev/dsk/c0t4d0s4	ufs	510.24	/opt

4. To return to the List menu, press F6.

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

1. Log in as superuser.
2. Type:

```
# lufslist
```

```
BE_name Specifies the name of the boot environment to view file system specifics
```

The following example displays a list.

Filesystem	fstype	size(Mb)	Mounted on
------------	--------	----------	------------

/dev/dsk/c0t0d0s1	swap	512.11	-
/dev/dsk/c0t4d0s3	ufs	3738.29	/
/dev/dsk/c0t4d0s4	ufs	510.24	/opt

Solaris Live Upgrade (Examples)

This chapter provides examples of creating a boot environment, then upgrading and activating it by using either a character user interface (CUI) or a command-line interface (CLI). Examples of switching back to the original boot environment are also provided.

Example of Upgrading With Solaris Live Upgrade (Command-Line Interface)

In this example, a new boot environment is created by using the `lucreate` command on a system that is running the Solaris 2.6 release. The new boot environment is upgraded to the Solaris 9 release by using the `luupgrade` command. The upgraded boot environment is activated by using the `luactivate` command. An example of falling back to the original boot environment is also given.

Install Live Upgrade on the Active Boot Environment

1. Insert the Solaris 9 DVD or Solaris 9 Software 2 of 2 CD.
2. Follow the step for the media you are using.
 - If you are using the Solaris 9 DVD, change directories to the installer and run the installer.

```
# cd /cdrom/cdrom0/Solaris_9/Tool/Installers  
# ./liveupgrade20
```

The Solaris Web Start installer is displayed.

- If you are using the Solaris 9 Software 2 of 2 CD, run the installer.

```
% ./installer
```

The Solaris Web Start installer is displayed.

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

Create a Boot Environment

The source boot environment is named `c0t4d0s0` by using the `-c` option. Naming the source boot environment is required the first time Solaris Live Upgrade is used on the system to create a boot environment. Any subsequent boot environment creations do not require using the `-c` option.

The new boot environment is named `c0t15d0s0`. The `-A` option creates a description that is associated with the boot environment name.

The root (`/`) file system is copied to the new boot environment. Also, a new swap slice is created rather than sharing the source boot environment's swap slice.

```
# lucreate -A 'BE_description' -c c0t4d0s0 -m /:/dev/dsk/c0t15d0s0:ufs \
-m -:/dev/dsk/c0t15d0s1:swap -n c0t15d0s0
```

Upgrade the Inactive Boot Environment

The inactive boot environment is named `c0t15d0s0`. The operating system image to be used for the upgrade is taken from the network.

```
# luupgrade -n c0t15d0s0 -u -s /net/ins3-svr/export/s9/combined.s9s_wos
```

Check If Boot Environment Is Bootable

The `lustatus` command reports if the boot environment creation is complete and if the boot environment is bootable.

```
# lustatus
```

```
BE_name                Complete  Active  ActiveOnReboot  CopyStatus
```

```
-----
c0t4d0s0          yes      yes      yes      -
c0t15d0s0        yes      no       no       -
```

Activate the Inactive Boot Environment

The c0t15d0s0 boot environment is made bootable with the `luactivate` command. The system is then rebooted and c0t15d0s0 becomes the active boot environment. The c0t4d0s0 boot environment is now inactive.

```
# luactivate c0t15d0s0
# init 6
```

Fall Back to the Source Boot Environment

Three procedures for falling back depend on your new boot environment activation situation:

- The activation is successful, but you want to return to the original boot environment. See “To Fall Back Despite Successful Boot Environment Creation” on page 309.
- The activation fails and you can boot back to the original boot environment. See “To Fall Back From a Failed Boot Environment Activation” on page 309.
- The activation fails and you must boot back to the original boot environment by using media or a net installation image. See “To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 310.

To Fall Back Despite Successful Boot Environment Creation

In this example, the original c0t4d0s0 boot environment is reinstated as the active boot environment although it was activated successfully. The device name is `first_disk`.

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

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.

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  
# umount /mnt  
# init 6
```

Example of Upgrading Using Solaris Live Upgrade (Character Interface)

In this example, a new boot environment is created on a system that is running the Solaris 2.6 release. The new boot environment is upgraded to the Solaris 9 release. The upgraded boot environment is then activated.

Install Live Upgrade on the Active Boot Environment

1. Insert the Solaris 9 DVD or Solaris 9 Software 2 of 2 CD.
2. Run the installer for the media you are using.
 - If you are using the Solaris 9 DVD, change directories to the installer and run the installer.

```
# cd /cdrom/cdrom0/Solaris_9/Tool/Installers  
# ./liveupgrade20
```

The Solaris Web Start installer is displayed.

- If you are using the Solaris 9 Software 2 of 2 CD, run the installer.

```
% ./installer
```

The Solaris Web Start installer displays.

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

Create a Boot Environment

In this example, the source boot environment is named `c0t4d0s0`. The root (`/`) file system is copied to the new boot environment. Also, a new swap slice is created rather than sharing the source boot environment's swap slice.

1. Display the character interface:

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

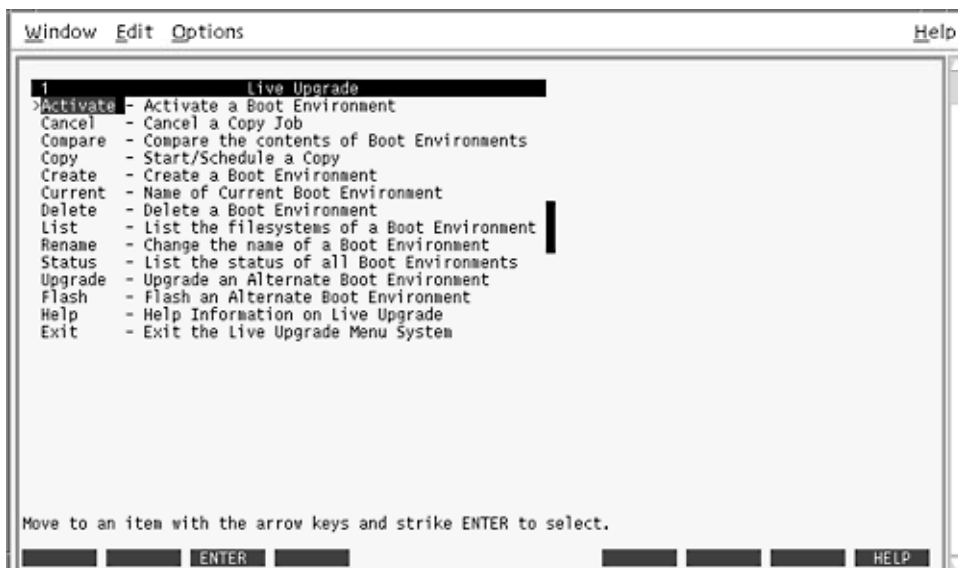


FIGURE 35-1 Solaris Live Upgrade Main Menu

2. From the main menu, select Create.

Name of Current Boot Environment: **c0t4d0s0**
Name of New Boot Environment: **c0t15d0s0**

3. Press F3.

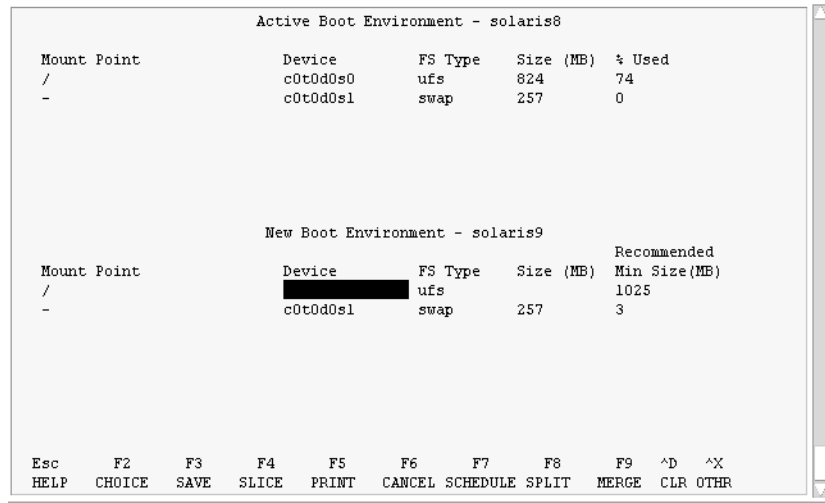


FIGURE 35-2 Solaris Live Upgrade Configuration Menu

4. From the Configuration menu, press F2 to display the Choices menu.
5. Choose slice 0 from disk c0t15d0 for root (/).
6. From the configuration menu, create a new slice for swap on c0t15d0 by selecting a swap slice to be split.
7. Press F2 to display the Choices menu.
8. Select slice 1 from disk c0t15d0 for the new swap slice.
9. Press F3 to create the new boot environment.

Upgrade the Inactive Boot Environment

The operating system image is taken from the network for the upgrade.

1. From the main menu, select Upgrade.

Name of New Boot Environment: **c0t15d0s0**
Package Media: /net/ins3-svr/export/s9/combined.s9s_wos

2. Press F3.

Activate the Inactive Boot Environment

The c0t15d0s0 boot environment is made bootable. The system is then rebooted and c0t15d0s0 becomes the active boot environment. The c0t4d0s0 boot environment is now inactive.

1. From the main menu, select Activate.

```
Name of Boot Environment: c0t15d0s0
Do you want to force a Live Upgrade sync operations: no
```

2. Press F3.
3. Press Return.
4. Type:

```
# init 6
```

If a fallback is necessary, use the command-line procedures in the previous example: “Fall Back to the Source Boot Environment” on page 309.

Solaris Live Upgrade (Command Reference)

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

TABLE 36-1 Solaris Live Upgrade Command-Line Options

Task	Command
Activate an inactive boot environment.	luactivate(1M)
Cancel a scheduled copy or create job.	lucancel(1M)
Compare an active boot environment with an inactive boot environment.	lucompare(1M)
Recopy file systems to update an inactive boot environment.	lucopy
Create a boot environment.	lucreate(1M)
Name the active boot environment.	lucurr(1M)
Delete a boot environment.	ludelete(1M)
Add a description to a boot environment name.	ludesc(1M)
List critical file systems for each boot environment.	lufslist(1M)
Enable a mount of all of the file systems in a boot environment. This command enables you to modify the files in a boot environment while that boot environment is inactive.	lumount(1M)
Rename a boot environment.	lurename(1M)
List status of all boot environments.	lustatus(1M)

TABLE 36-1 Solaris Live Upgrade Command-Line Options (Continued)

Task	Command
Enable an unmount of all the file systems in a boot environment. This command enables you to modify the files in a boot environment while that boot environment is inactive.	luumount(1M)
Upgrade an operating environment or install a flash archive on an inactive boot environment.	luupgrade(1M)

Solaris Software Reference (Topics)

This section provides reference information about the Solaris CDs, locales, and packages.

Chapter 38	Describes the primary DVD and CDs that are included in the media kits for Solaris 9.
Chapter 39	Contains a list of the platform names and groups of various hardware platforms.
Chapter 40	Contains a list of the values that are needed to set the <code>locale</code> keyword in a profile or to preconfigure a locale.

Organization of Solaris 9 Media (Reference)

This chapter describes the primary DVD and CD media that are included in the media kits for Solaris 9 software.

Solaris 9 Media

The following tables list the primary DVD and CD media for the Solaris 9 release. You receive a multilingual media kit that contains DVD or CD media for English, other languages, and locale software.

TABLE 38-1 Multilingual DVD Media

DVD Title	Description
Solaris 9 DVD	Contains the following: <ul style="list-style-type: none">■ Software, tools, and configuration information to install the Solaris product and all partial locales■ ExtraValue software that includes supported and unsupported software■ Installers■ Source code for some third-party public domain software■ Interface software and documentation that have been localized■ The Solaris documentation set for English, European, and Asian languages that includes Japanese For directory structures, see Figure 38-1.

TABLE 38-2 Multilingual CD Media

CD Title	Description
You receive one of these CDs: <ul style="list-style-type: none"> ■ Solaris 9 Installation CD ■ Solaris 9 Multilingual Installation CD 	<ul style="list-style-type: none"> ■ Contains scripts to install Solaris software. For directory structures, see Figure 38-2. ■ Contains scripts to install Solaris software and all partial locales. For directory structures, see Figure 38-2.
Solaris 9 Software 1 of 2 CD	Contains the software, tools, and configuration information to install the Solaris product. For directory structures, see Figure 38-3.
Solaris 9 Software 2 of 2 CD	<p>Contains the following:</p> <ul style="list-style-type: none"> ■ A limited number of packages, which the software prompts you to install if necessary ■ ExtraValue software that includes supported and unsupported software ■ Installers ■ Source code for some third-party public domain software <p>For directory structures, see Figure 38-4.</p> <p>The installation program prompts you for this CD if necessary.</p>
Solaris 9 Languages CD	<p>Contains interface software and documentation that has been localized. For directory structures, see Figure 38-5.</p> <p>The installation program prompts you for this CD if necessary to support languages for specific geographic regions.</p>
Solaris 9 Documentation 1 of 2 CD	Contains the Solaris documentation set for English and European languages.
Solaris 9 Documentation 2 of 2 CD	Contains the Solaris documentation set for Asian languages that includes Japanese.

Directory Organization of Solaris 9 Media

This section describes top-level directories on each DVD and CD.

Solaris 9 DVD Directory Structure

The following figure shows the directory structure on the Solaris 9 DVD .

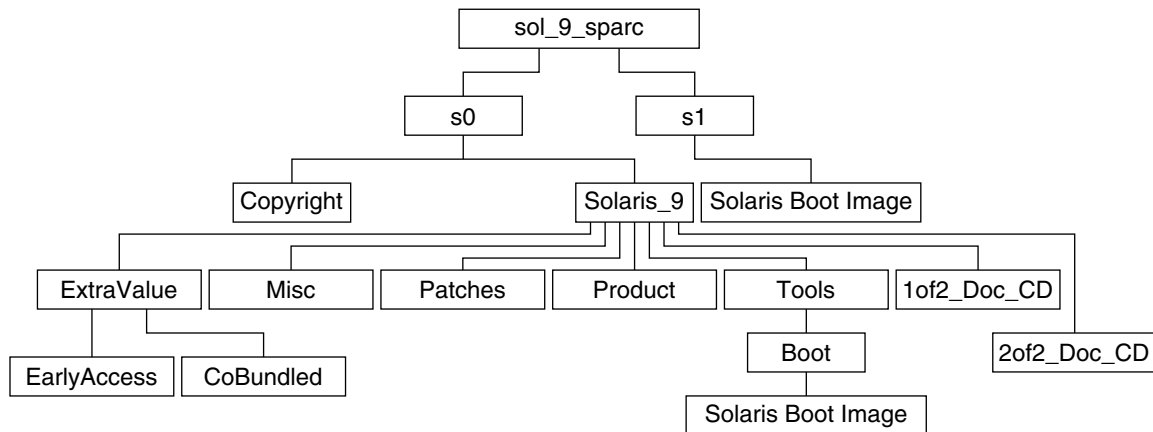


FIGURE 38-1 Solaris 9 DVD

- Slice 0 (s0) contains the Copyright and Solaris_9 directories. The Solaris_9 directory contains all the tools, software, and configuration information necessary to install the Solaris 9 software product. Slice 0 contains the following directories:
 - ExtraValue – Two subdirectories:
 - CoBundled – Fully supported Solaris products not directly part of the Solaris operating environment that includes Solaris Web Start Wizards SDK 3.0.1 and SunScreen 3.2
 - EarlyAccess – Preliminary evaluation software
 - Misc – The jumpstart_sample directory, which includes a rules file, a check script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
 - Patches – All the Solaris 9 patches available at the time of this release.
 - Product – The Solaris 9 packages and control files.
 - Tools – The Solaris 9 installation tools, which include the following:
 - The Boot subdirectory, which contains the Solaris 9 miniroot.
 - The scripts add_install_client, dial, rm_install_client, and setup_install_server.
 - An Installer subdirectory that contains an installer for Solaris Live Upgrade. Solaris Live Upgrade is automatically installed when you install the Solaris software. However, in order to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see “To Install Solaris Live Upgrade” on page 249.

- 1of2_Doc_CD – Documentation for English and European Languages
- 2of2_Doc_CD – Documentation for Asian Languages
- Slice 1 (s1) contains the Solaris 9 miniroot.

Solaris 9 Installation CD Directory Structure

The following figure shows the directory structure on the Solaris 9 Multilingual Installation CD or Solaris 9 Installation CD.

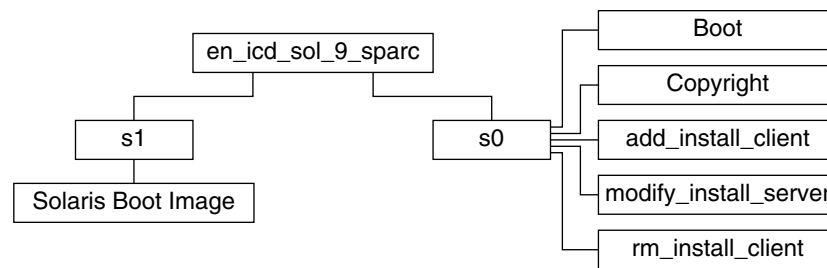


FIGURE 38–2 Solaris 9 Installation CD

- Slice 0 (s0) contains scripts that install the Solaris software and a Boot subdirectory that contains the Solaris miniroot. These scripts include the following:
 - `add_install_client`
 - `modify_install_server`
 - `rm_install_client`
- Slice 1 (s1) contains the Solaris 9 miniroot.

Solaris 9 Software CDs Directory Structures

The following figures show the directory structure on the Solaris 9 Software CDs.

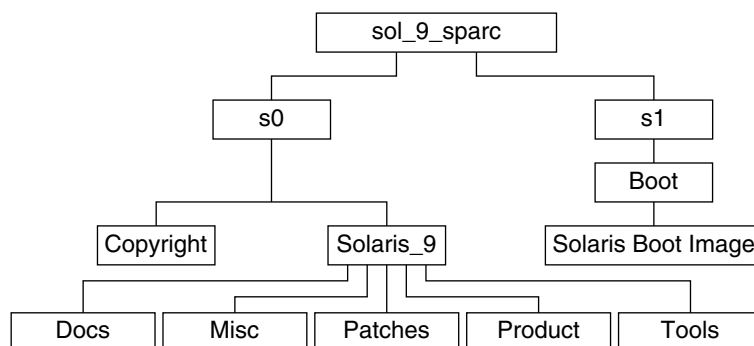


FIGURE 38-3 Solaris 9 Software 1 of 2 CD

- Slice 0 (s0) contains the `Copyright` and `Solaris_9` directories. The `Solaris_9` directory contains all the tools, software, and configuration information necessary to install, at a minimum, the Solaris 9 software product, including the Core Solaris Software and End User Solaris Software groups. Slice 0 contains the following directories:
 - `Docs` – An empty directory.
 - `Misc` – The `jumpstart_sample` directory, which includes a rules file, a check script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
 - `Patches` – All the Solaris 9 patches available at the time the Solaris 9 Software 1 of 2 CD was created.
 - `Product` – The Solaris 9 packages and control files.
 - `Tools` – The Solaris 9 installation tools, which include `add_install_client`, `dial`, `rm_install_client`, and `setup_install_server`.
- Slice 1 (s1) contains the Solaris 9 miniroot.

The following figure shows the directories on the Solaris 9 Software 2 of 2 CD.

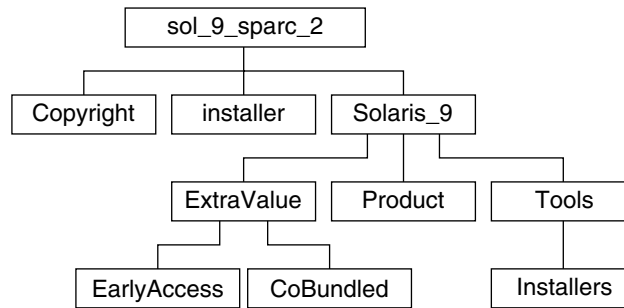


FIGURE 38-4 Solaris 9 Software 2 of 2 CD

The `sol_9_sparc_2` directory contains `Copyright`, Solaris Web Start installer, and `Solaris_9` directories. The `Solaris_9` directory contains the following:

- `ExtraValue` – Two subdirectories:
 - `CoBundled` – Fully supported Solaris products not directly part of the Solaris operating environment that includes Solaris Web Start Wizards SDK 3.0.1 and SunScreen 3.2
 - `EarlyAccess` – Preliminary evaluation software
- `Product` – A limited number of packages, Developer Solaris Software, Entire Solaris Software, and Entire Solaris Software Plus OEM Support Groups
- `Tools` – Solaris 9 installation tools that include the following:
 - The `add_to_install_server` script.
 - An `Installer` subdirectory that contains an installer for Solaris Live Upgrade. Solaris Live Upgrade is automatically installed when you install the Solaris software, but to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see “To Install Solaris Live Upgrade” on page 249.

Solaris 9 Languages CD Directory Structure

The following figure shows the directory structure of the Solaris 9 Languages CD.

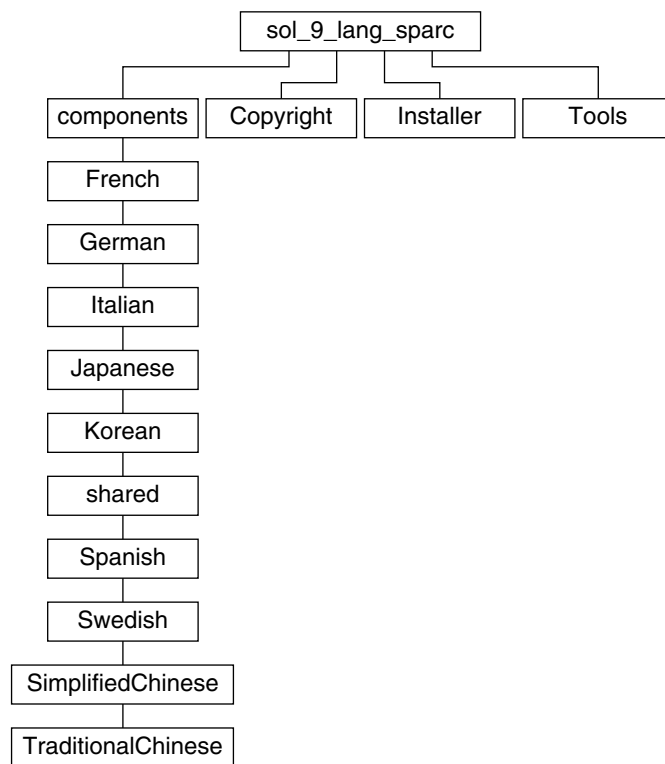


FIGURE 38-5 Solaris 9 Languages CD

The `sol_9_lang_sparc` directory contains the Solaris Web Start installer, which installs the Solaris language and locale software. The `sol_9_lang_sparc` directory contains these directories:

- `components` – SimplifiedChinese, French, German, Italian, Japanese, Korean, Spanish, Swedish, and TraditionalChinese locale packages and packages that are shared by all locales
- `Copyright` – Copyright page
- `installer` – Solaris Web Start installer
- `Tools` – The `add_to_install_server` script for creating an install server

For more information on locale software, see *International Language Environments Guide*.

Platform Names and Groups (Reference)

You need to know your system architecture (platform group) if you are adding clients for a network installation and the platform name if you are writing a custom JumpStart installation rules file.

Some examples of platform names and groups follow. For a full list of systems, see *Solaris 9 Sun Hardware Platform Guide*.

TABLE 39-1 Example of Platform Names and Groups

System	Platform Name	Platform Group
Ultra 5	SUNW,Ultra-5_10	sun4u
SPARCstation 20	SUNW,SPARCstation-20	sun4m

Note – On a running system, you can also use the `uname -i` command to determine a system's *platform name* or the `uname -m` command to determine a system's *platform group*.

Locale Values (Reference)

The tables below list the values that are needed to set the `locale` keyword in a profile or to preconfigure a locale.

A *locale* determines how online information is displayed in a specific language and region. A language might also include more than one locale to accommodate regional differences, such as differences in the format of date and time, numeric and monetary conventions, and spelling. For additional information about locales, see the *International Language Environments Guide*.

TABLE 40-1 Asia

Locale	User Interface	Territory	Codeset	Language Support
hi_IN.UTF-8	English	India	UTF-8 ¹	Hindi (UTF-8) Unicode 3.1
ja	Japanese	Japan	eucJP ²	Japanese (EUC) JIS X 0201-1976 JIS X 0208-1990 JIS X 0212-1990
ja_JP.eucJP	Japanese	Japan	eucJP	Japanese (EUC) JIS X 0201-1976 JIS X 0208-1990 JIS X 0212-1990
ja_JP.PCK	Japanese	Japan	PCK ³	Japanese (PC kanji) JIS X 0201-1976

¹UTF-8 is the UTF-8 defined in ISO/IEC 10646-1:2000 and also Unicode 3.1.

²eucJP signifies the Japanese EUC codeset. Specification of `ja_JP.eucJP` locale conforms to UI_OSF Japanese Environment Implementation Agreement Version 1.1 and `ja` locale conforms to the traditional specification from the past Solaris releases.

³PCK is also known as Shift_JIS (SJIS).

TABLE 40-1 Asia (Continued)

Locale	User Interface	Territory	Codeset	Language Support
				JIS X 0208-1990
ja_JP.UTF-8	Japanese	Japan	UTF-8	Japanese (UTF-8) Unicode 3.1
ko_KR.EUC	Korean	Korea	1001	Korean (EUC) KS X 1001
ko_KR.UTF-8	Korean	Korea	UTF-8	Korean (UTF-8) Unicode 3.1
th_TH.UTF-8	English	Thailand	UTF-8	Thai (UTF-8) Unicode 3.1
th_TH.TIS620	English	Thailand	TIS620.2533	Thai TIS620.2533
zh_CN.EUC	Simplified Chinese	PRC	gb2312 ⁴	Simplified Chinese (EUC) GB2312-1980
zh_CN.GBK	Simplified Chinese	PRC	GBK ⁵	Simplified Chinese (GBK)
zh_CN.GB18030	Simplified Chinese	PRC	GB18030-2000	Simplified Chinese (GB18030-2000) GB18030-2000
zh_CN.UTF-8	Simplified Chinese	PRC	UTF-8	Simplified Chinese (UTF-8) Unicode 3.1
zh_HK.BIG5HK	Traditional Chinese	Hong Kong	Big5+HKSCS	Traditional Chinese (BIG5+HKSCS)
zh_HK.UTF-8	Traditional Chinese	Hong Kong	UTF-8	Traditional Chinese (UTF-8) Unicode 3.1
zh_TW.EUC	Traditional Chinese	Taiwan	cns11643	Traditional Chinese (EUC) CNS 11643-1992
zh_TW.BIG5	Traditional Chinese	Taiwan	BIG5	Traditional Chinese (BIG5)
zh_TW.UTF-8	Traditional Chinese	Taiwan	UTF-8	Traditional Chinese (UTF-8) Unicode 3.1

⁴gb2312 signifies Simplified Chinese EUC codeset, which contains GB 1988-80 and GB 2312-80.

⁵GBK signifies GB extensions. This includes all GB 2312-80 characters and all Unified Han characters of ISO/IEC 10646-1, as well as Japanese Hiragana and Katakana characters. It also includes many characters of Chinese, Japanese, and Korean character sets and of ISO/IEC 10646-1.

TABLE 40-2 Australasia

Locale	User Interface	Territory	Codeset	Language Support
en_AU.ISO8859-1	English	Australia	ISO8859-1	English (Australia)
en_NZ.ISO8859-1	English	New Zealand	ISO8859-1	English (New Zealand)

TABLE 40-3 Central America

Locale	User Interface	Territory	Codeset	Language Support
es_CR.ISO8859-1	Spanish	Costa Rica	ISO8859-1	Spanish (Costa Rica)
es_GT.ISO8859-1	Spanish	Guatemala	ISO8859-1	Spanish (Guatemala)
es_NI.ISO8859-1	Spanish	Nicaragua	ISO8859-1	Spanish (Nicaragua)
es_PA.ISO8859-1	Spanish	Panama	ISO8859-1	Spanish (Panama)
es_SV.ISO8859-1	Spanish	El Salvador	ISO8859-1	Spanish (El Salvador)

TABLE 40-4 Central Europe

Locale	User Interface	Territory	Codeset	Language Support
cs_CZ.ISO8859-2	English	Czech Republic	ISO8859-2	Czech (Czech Republic)
de_AT.ISO8859-1	German	Austria	ISO8859-1	German (Austria)
de_AT.ISO8859-15	German	Austria	ISO8859-15	German (Austria, ISO8859-15 - Euro)
de_CH.ISO8859-1	German	Switzerland	ISO8859-1	German (Switzerland)
de_DE.UTF-8	German	Germany	UTF-8	German (Germany, Unicode 3.1)
de_DE.ISO8859-1	German	Germany	ISO8859-1	German (Germany)
de_DE.ISO8859-15	German	Germany	ISO8859-15	German (Germany, ISO8859-15 - Euro)
fr_CH.ISO8859-1	French	Switzerland	ISO8859-1	French (Switzerland)
hu_HU.ISO8859-2	English	Hungary	ISO8859-2	Hungarian (Hungary)
pl_PL.ISO8859-2	English	Poland	ISO8859-2	Polish (Poland)
pl_PL.UTF-8	English	Poland	UTF-8	Polish (Poland, Unicode 3.1)
sk_SK.ISO8859-2	English	Slovakia	ISO8859-2	Slovak (Slovakia)

TABLE 40-5 Eastern Europe

Locale	User Interface	Territory	Codeset	Language Support
bg_BG.ISO8859-5	English	Bulgaria	ISO8859-5	Bulgarian (Bulgaria)
et_EE.ISO8859-15	English	Estonia	ISO8859-15	Estonian (Estonia)

TABLE 40-5 Eastern Europe (Continued)

Locale	User Interface	Territory	Codeset	Language Support
hr_HR.ISO8859-2	English	Croatia	ISO8859-2	Croatian (Croatia)
lt_LT.ISO8859-13	English	Lithuania	ISO8859-13	Lithuanian (Lithuania)
lv_LV.ISO8859-13	English	Latvia	ISO8859-13	Latvian (Latvia)
mk_MK.ISO8859-5	English	Macedonia	ISO8859-5	Macedonian (Macedonia)
ro_RO.ISO8859-2	English	Romania	ISO8859-2	Romanian (Romania)
ru_RU.KOI8-R	English	Russia	KOI8-R	Russian (Russia, KOI8-R)
ru_RU.ANSI1251	English	Russia	ansi-1251	Russian (Russia, ANSI 1251)
ru_RU.ISO8859-5	English	Russia	ISO8859-5	Russian (Russia)
ru_RU.UTF-8	English	Russia	UTF-8	Russian (Russia, Unicode 3.1)
sh_BA.ISO8859-2@bosnia	English	Bosnia	ISO8859-2	Bosnian (Bosnia)
sl_SI.ISO8859-2	English	Slovenia	ISO8859-2	Slovenian (Slovenia)
sq_AL.ISO8859-2	English	Albania	ISO8859-2	Albanian (Albania)
sr_YU.ISO8859-5	English	Serbia	ISO8859-5	Serbian (Serbia)
tr_TR.ISO8859-9	English	Turkey	ISO8859-9	Turkish (Turkey)
tr_TR.UTF-8	English	Turkey	UTF-8	Turkish (Turkey, Unicode 3.1)

TABLE 40-6 Middle East

Locale	User Interface	Territory	Codeset	Language Support
He	English	Israel	ISO8859-8	Hebrew (Israel)

TABLE 40-7 North Africa

Locale	User Interface	Territory	Codeset	Language Support
ar_EG.UTF-8	English	Egypt	UTF-8	Arabic (Egypt)

TABLE 40-7 North Africa (Continued)

Locale	User Interface	Territory	Codeset	Language Support
Ar	English	Egypt	ISO8859-6	Arabic (Egypt)

TABLE 40-8 North America

Locale	User Interface	Territory	Codeset	Language Support
en_CA.ISO8859-1	English	Canada	ISO8859-1	English (Canada)
en_US.ISO8859-1	English	USA	ISO8859-1	English (U.S.A.)
en_US.ISO8859-15	English	USA	ISO8859-15	English (U.S.A., ISO8859-15 - Euro)
en_US.UTF-8	English	USA	UTF-8	English (U.S.A., Unicode 3.1)
fr_CA.ISO8859-1	French	Canada	ISO8859-1	French (Canada)
es_MX.ISO8859-1	Spanish	Mexico	ISO8859-1	Spanish (Mexico)

TABLE 40-9 Northern Europe

Locale	User Interface	Territory	Codeset	Language Support
da_DK.ISO8859-1	English	Denmark	ISO8859-1	Danish (Denmark)
da_DK.ISO8859-15	English	Denmark	ISO8859-15	Danish (Denmark, ISO8859-15-Euro)
fi_FI.ISO8859-1	English	Finland	ISO8859-1	Finnish (Unicode 3.1)
fi_FI.ISO8859-15	English	Finland	ISO8859-15	Finnish (Finland, ISO8859-15-Euro)
fi_FI.UTF-8	English	Finland	UTF-8	Finnish (Finland)
is_IS.ISO8859-1	English	Iceland	ISO8859-1	Icelandic (Iceland)
no_NO.ISO8859-1@bokmal	English	Norway	ISO8859-1	Norwegian (Norway-Bokmal)
no_NO.ISO8859-1@nyorsk	English	Norway	ISO8859-1	Norwegian (Norway-Nynorsk)
sv_SE.ISO8859-1	Swedish	Sweden	ISO8859-1	Swedish (Sweden)
sv_SE.ISO8859-15	Swedish	Sweden	ISO8859-15	Swedish (Sweden, ISO8859-15-Euro)
sv_SE.UTF-8	Swedish	Sweden	UTF-8	Swedish (Sweden, Unicode 3.1)

TABLE 40-10 South America

Locale	User Interface	Territory	Codeset	Language Support
es_AR.ISO8859-1	Spanish	Argentina	ISO8859-1	Spanish (Argentina)
es_BO.ISO8859-1	Spanish	Bolivia	ISO8859-1	Spanish (Bolivia)
es_CL.ISO8859-1	Spanish	Chile	ISO8859-1	Spanish (Chile)
es_CO.ISO8859-1	Spanish	Colombia	ISO8859-1	Spanish (Colombia)
es_EC.ISO8859-1	Spanish	Ecuador	ISO8859-1	Spanish (Ecuador)
es_PE.ISO8859-1	Spanish	Peru	ISO8859-1	Spanish (Peru)
es_PY.ISO8859-1	Spanish	Paraguay	ISO8859-1	Spanish (Paraguay)
es_UY.ISO8859-1	Spanish	Uruguay	ISO8859-1	Spanish (Uruguay)
es_VE.ISO8859-1	Spanish	Venezuela	ISO8859-1	Spanish (Venezuela)
pt_BR.ISO8859-1	English	Brazil	ISO8859-1	Portuguese (Brazil)
pt_BR.UTF-8	English	Brazil	UTF-8	Portuguese (Brazil, Unicode 3.1)

TABLE 40-11 South Europe

Locale	User Interface	Territory	Codeset	Language Support
ca_ES.ISO8859-1	English	Spain	ISO8859-1	Catalan (Spain)
ca_ES.ISO8859-15	English	Spain	ISO8859-15	Catalan (Spain, ISO8859-15 - Euro)
e1_GR.ISO8859-7	English	Greece	ISO8859-7	Greek (Greece)
es_ES.ISO8859-1	Spanish	Spain	ISO8859-1	Spanish (Spain)
es_ES.ISO8859-15	Spanish	Spain	ISO8859-15	Spanish (Spain, ISO8859-15 - Euro)
es_ES.UTF-8	Spanish	Spain	UTF-8	Spanish (Spain, Unicode 3.1)
it_IT.ISO8859-1	Italian	Italy	ISO8859-1	Italian (Italy)
it_IT.ISO8859-15	Italian	Italy	ISO8859-15	Italian (Italy, ISO8859-15 - Euro)
it_IT.UTF-8	Italian	Italy	UTF-8	Italian (Italy, Unicode 3.1)
pt_PT.ISO8859-1	English	Portugal	ISO8859-1	Portuguese (Portugal)
pt_PT.ISO8859-15	English	Portugal	ISO8859-15	Portuguese (Portugal, ISO8859-15 - Euro)

TABLE 40–12 Western Europe

Locale	User Interface	Territory	Codeset	Language Support
en_GB.ISO8859-1	English	Great Britain	ISO8859-1	English (Great Britain)
en_IE.ISO8859-1	English	Ireland	ISO8859-1	English (Ireland)
fr_BE.ISO8859-1	French	Belgium- Walloon	ISO8859-1	French (Belgium-Walloon, Unicode 3.1)
fr_BE.UTF-8	French	Belgium- Walloon	UTF-8	French (Belgium-Walloon, Unicode 3.1)
fr_FR.ISO8859-1	French	France	ISO8859-1	French (France)
fr_FR.UTF-8	French	France	UTF-8	French (France, Unicode 3.1)
nl_BE.ISO8859-1	English	Belgium- Flemish	ISO8859-1	Dutch (Belgium-Flemish)
nl_NL.ISO8859-1	English	Netherlands	ISO8859-1	Dutch (Netherlands)

Troubleshooting (Tasks)

This chapter contains a list of specific error messages and general problems you might encounter when installing Solaris 9 software. The chapter also explains how to fix the problems. Start by using this list of sections to determine where in the installation process the problem occurred.

- “Problems Setting Up Network Installations” on page 337
- “Problems With Booting a System” on page 338
- “Initial Installation of the Solaris 9 Operating Environment” on page 342
- “Upgrading the Solaris 9 Operating Environment” on page 342

Note – When you see the phrase “bootable media,” this means one of the installation programs: Solaris `suninstall` program, Solaris Web Start program, or custom JumpStart.

Problems Setting Up Network Installations

Unknown client “*host_name*”

Cause: The *host_name* argument in the `add_install_client` command is not a host in the name service.

Add the host *host_name* to the name service and execute the `add_install_client` command again.

Problems With Booting a System

Booting From Media, Error Messages

le0: No carrier - transceiver cable problem

Cause: The system is not connected to the network.

Solution: If this is a non-networked system, ignore this message. If this is a networked system, ensure that the Ethernet cabling is attached securely.

The file just loaded does not appear to be executable

Cause: The system cannot find the proper media for booting.

Solution: Verify that the system has been set up properly to install the Solaris 9 software from the network from an install server. For example, ensure that you specified the correct platform group for the system when you set it up.

Or, if you did not copy the images of the Solaris 9 DVD or Solaris 9 Software 1 of 2, Solaris 9 Software 2 of 2, and Solaris 9 Languages CDs to the install server, ensure the Solaris 9 DVD or Solaris 9 Software 1 of 2 CD is mounted and accessible on the install server.

boot: cannot open /kernel/unix

Cause: This error occurs when you override the location of the boot - file by explicitly setting it to /kernel/unix.

Solution:

- Reset the boot -file in the PROM to “ ” (blank).
- Ensure that the diag-switch is set to off and to true.

Can't boot from file/device

Cause: The installation media cannot find the bootable media.

Solution: Ensure that the following conditions are met:

- The DVD-ROM or CD-ROM drive is installed properly and turned on.
- Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD is inserted into the drive.
- The disc is free of damage or dirt.

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

Description: This is an informational message.

Solution: Ignore the message and continue with the installation.

Booting From Media, General Problems

The system does not boot.

Description: When initially setting up a custom JumpStart server, you might encounter boot problems that do not return an error message. To verify information about the system and how the system is booting, run the boot command with the `-v` option. When you use the `-v` option, the boot command displays verbose debugging information on the screen.

Note – If this flag is not given, the messages are still printed, but the output is directed to the system logfile. For more information, see `syslogd(1M)`.

Solution: At the `ok` prompt, type the following: `ok boot net -v - install`.

Boot from DVD media fails on systems with Toshiba SD-M 1401 DVD-ROM

If your system has a Toshiba SD-M1401 DVD-ROM with firmware revision 1007, the system cannot boot from the Solaris 9 DVD.

Solution: Apply patch 111649-03, or later version, to update the Toshiba SD-M1401 DVD-ROM drive's firmware. Patch 111649-03 is included on the Solaris 9 Supplement CD.

Booting From the Network, Error Messages

WARNING: getfile: RPC failed: error 5 (RPC Timed out).

Description: This error occurs when you have two or more servers on a network responding to an install client's boot request. The install client connects to the wrong boot server, and the installation hangs. The following specific reasons might cause this error to occur:

Cause: *Reason 1:* `/etc/bootparams` files might exist on different servers with an entry for this install client.

Solution: *Reason 1:* Ensure that servers on the network do not have multiple `/etc/bootparams` entries for the install client. If they do have multiple entries, remove duplicate client entries in the `/etc/bootparams` file on all install servers and boot servers except the one you want the install client to use.

Cause: *Reason 2:* Multiple `/tftpboot` or `/rplboot` directory entries might exist for this install client.

Solution: *Reason 2:* Ensure that servers on the network do not have multiple /tftpboot or /rplboot directory entries for the install client. If they do have multiple entries, remove duplicate client entries from the /tftpboot or /rplboot directories on all install servers and boot servers except the one you want the install client to use.

Cause: *Reason 3:* An install client entry might exist in the /etc/bootparams file on a server and an entry in another /etc/bootparams file that enables all systems to access the profile server. Such an entry resembles the following:

```
* install_config=profile_server:path
```

A line that resembles the previous entry in the NIS or NIS+ bootparams table can also cause this error.

Solution: *Reason 3:* If a wildcard entry is in the name service bootparams map or table (for example, * install_config=), delete it and add it to the /etc/bootparams file on the boot server.

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

Cause: This error occurs on a system that you are attempting to install from the network. The system is not set up correctly.

Solution: Ensure that you correctly set up the system to install from the network. See “Adding Systems to Be Installed From the Network” on page 87.

prom_panic: Could not mount file system

Cause: This error occurs when you are installing Solaris 9 from a network, but the boot software cannot locate the following:

- Solaris 9 DVD, either the DVD or a copy of the DVD image on the install server
- Solaris 9 Software 1 of 2 CD image, either the Solaris 9 Software 1 of 2 CD or a copy of the Solaris 9 Software 1 of 2 CD image on the install server

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

- If you are installing Solaris 9 from the install server’s DVD-ROM or CD-ROM drive, ensure that the Solaris 9 DVD or Solaris 9 Software 1 of 2 is inserted in the CD-ROM drive, is mounted, and is shared in the /etc/dfs/dfstab file.
- If installing from a copy of the Solaris 9 DVD image or Solaris 9 Software 1 of 2 CD image on the install server’s disk, ensure that the directory path to the copy is shared in the /etc/dfs/dfstab file.

See the man page, install_server.

Timeout waiting for ARP/RARP packet...

Cause: *Reason 1:* The client is trying to boot from the network, but it cannot find a system that knows about the client.

Solution: *Reason 1:* Verify the system's host name is in the NIS or NIS+ name service. Also, verify the bootparams search order in the boot server's `/etc/nsswitch.conf` file.

For example, the following line in the `/etc/nsswitch.conf` file indicates that JumpStart or the Solaris `suninstall` program first looks in the NIS maps for bootparams information. If the program does not find any information, the JumpStart program or the Solaris `suninstall` program looks in the boot server's `/etc/bootparams` file.

```
bootparams: nis files
```

Cause: *Reason 2:* The client's Ethernet address is not correct.

Solution: *Reason 2:* Verify that the client's Ethernet address in the install server's `/etc/ethers` file is correct.

Cause: *Reason 3:* In a custom JumpStart installation, the `add_install_client` command specifies the platform group that uses a specified server as an install server. If the wrong architecture value is used when using the `add_install_client`, this problem occurs. For example, the machine you want to install is a sun4u, but you used sun4m instead.

Solution: *Reason 3:* Rerun `add_install_client` with the correct architecture value.

Booting From the Network, General Problems

The system boots from the network, but from a system other than the specified install server.

Cause: An `/etc/bootparams` and perhaps `/etc/ethers` entry exist on another system for the client.

Solution: On the name server, update the `/etc/bootparams` entry for the system that is being installed. The entry should conform to the following syntax:

```
install_system root=boot_server:path install=install_server:path
```

Also, ensure that only one bootparams entry is on the subnet for the install client.

After you set up an install server and configure the system to install Solaris 9 from the network, the system still does not boot.

Cause: The `tftpd` might not be running on the install server.

Solution: Be sure the `tftpd` daemon is running on the install server. Type the following command:

```
# ps -ef | grep tftpd
```

If this command does not return a line that indicates that the `tftpd` daemon is running, edit the `/etc/inetd.conf` file and remove the comment (`#`) character from the following line:

```
# tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd \  
-s /tftpboot
```

After making this change, try booting the system again.

Initial Installation of the Solaris 9 Operating Environment

Initial installation fails

Solution: If the Solaris installation fails, you must restart the installation. To restart the installation, boot the system from the Solaris 9 DVD, Solaris 9 Installation CD, the Solaris 9 Software 1 of 2, or from the network.

You cannot uninstall the Solaris software after the software has been partially installed. You must restore your system from a backup or begin the Solaris installation process again.

`/cdrom/Solaris_9/SUNWxxx/reloc.cpio: Broken pipe`

Description: This error message does not affect the installation.

Solution: Ignore the message and continue with the installation.

Upgrading the Solaris 9 Operating Environment

Upgrading, Error Messages

No upgradable disks

Cause: A swap entry in the `/etc/vfstab` file is causing the upgrade to fail.

Solution: Comment out the following lines in the `/etc/vfstab` file:

- All swap files and slices on disks not being upgraded

- Swap files that are no longer present
- Any unused swap slices

usr/bin/bzczt not found

Cause: Solaris Live Upgrade fails because of needing a patch cluster.

Solution: A patch is needed to install Solaris Live Upgrade. Go to <http://sunsolve.sun.com> for the patch.

Upgrading, General Problems

The upgrade option is not presented even though there is a version of Solaris software that's upgradable on the system.

Cause: *Reason 1:* The `/var/sadm` directory is a symlink or it is mounted from another file system.

Solution: *Reason 1:* Move the `/var/sadm` directory into the root (`/`) or `/var` file system.

Cause: *Reason 2:* The `/var/sadm/softinfo/INST_RELEASE` file is missing.

Solution: *Reason 2:* Create a new `INST_RELEASE` file by using the following template:

```
OS=Solaris
VERSION=x
REV=0
```

`x` Is the version of Solaris software on the system

Cause: *Reason 3:* `SUNWusr` is missing from `/var/sadm/softinfo`.

Solution: *Solution 3:* You need to do an initial installation. The Solaris software is not upgradable.

Couldn't shut down or initialize the md driver

Solution:

- If not a mirror, comment out in the `vsftab` file.
- If a mirror, break the mirror and reinstall.

The upgrade fails because the Solaris installation program cannot mount a file system.

Cause: During an upgrade, the script attempts to mount all the file systems that are listed in the system's `/etc/vfstab` file on the root (`/`) file system that is being upgraded. If the installation script cannot mount a file system, it fails and exits.

Ensure that all file systems in the system's `/etc/vfstab` file can be mounted. Comment out any file systems in the `/etc/vfstab` file that cannot be mounted or that might cause the problem so that the Solaris `suninstall` program does not try to mount them during the upgrade. Any system-based file systems that contain software to be upgraded (for example, `/usr`) cannot be commented out.

The upgrade fails

Description: The system does not have enough space for the upgrade.

Cause: Check Chapter 5 for the space problem and see if you can fix it without using auto-layout to reallocate space.

Problems upgrading mirrored roots

Solution: If you have problems upgrading when using Solaris Volume Manager mirrored roots, see "Solving Problems Related to Solaris Volume Manager" in *Solaris Volume Manager Administration Guide*.

▼ To Continue Upgrading After a Failed Upgrade

The upgrade fails and the system cannot be soft-booted. The failure is for reasons beyond your control, such as a power failure or a network connection failure.

1. **Reboot the system from the Solaris 9 DVD, Solaris 9 Installation CD, the Solaris 9 Software 1 of 2 CD, or from the network.**
2. **Choose the upgrade option for installation.**

The Solaris installation program determines if the system has been partially upgraded and continues the upgrade.

System Panics When Upgrading With Solaris Live Upgrade Running Veritas VxVm

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

1. **Create an inactive boot environment. See "Creating a New Boot Environment" on page 251.**
2. **Before upgrading the inactive boot environment, you must disable the existing Veritas software on the inactive boot environment.**
 - a. **Mount the inactive boot environment.**

```
# lumount inactive_boot_environment_name mount_point
```


For example:

```
# lumount solaris8 /.alt.12345
```

- b. Change to the directory that contains the `vfstab`, for example:**

```
# cd /.alt.12345/etc
```

- c. Make a copy of the inactive boot environment's `vfstab` file, for example:**

```
# cp vfstab vfstab.501
```

- d. In the copied `vfstab`, comment out all Veritas file system entries, for example:**

```
# sed vfstab.novxfs > vfstab < '/vx\|dsk/s/^\|#/g'
```

The first character of each line is changed to #, which makes the line a comment line. Note that this comment line is different than the system file comment lines.

- e. Copy the changed `vfstab` file, for example:**

```
# cp vfstab.novxfs vfstab
```

- f. Change directories to the inactive boot environment's system file, for example:**

```
# cd /.alt.12345/etc
```

- g. Make a copy of the inactive boot environment's system file, for example:**

```
# cp system system.501
```

- h. Comment out all "forceload:" entries that include `drv/vx`.**

```
# sed '/forceload: system.novxfs > system < drv\|vx/s/^\|*/'
```

The first character of each line is changed to *, which makes the line a command line. Note that this comment line is different than the `vfstab` file comment lines.

- i. Change directories to the `install-db` file on the inactive boot environment, for example:**

```
# cd /.alt.12345/etc
```

- j. Create the Veritas `install-db` file, for example:**

```
# touch vx/reconfig.d/state.d/install-db
```

- k. Unmount the inactive boot environment.**

```
# lumount inactive_boot_environment_name mount_point
```

- 3. Upgrade the inactive boot environment. See Chapter 33.**

- 4. Activate the inactive boot environment. See "Activating a Boot Environment" on page 282.**

- 5. Shut down the system.**

```
# init 0
```

6. Boot the inactive boot environment in single-user mode:

```
OK boot -s
```

Several messages and error messages that contain “vxvm” or “VXVM” are displayed that can be ignored. The inactive boot environment becomes active.

7. Upgrade Veritas.

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

```
# pkgrm VRTSvmsa
```

b. Change directories to the Veritas packages.

```
# cd /location_of_Veritas_software
```

c. Add the latest Veritas packages to the system:

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

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

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

9. Reboot the system.

```
# init 6
```

Installing or Upgrading Remotely (Tasks)

This appendix describes how to use the Solaris Web Start program to install or upgrade to the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive.

Note – If you are installing or upgrading the Solaris operating environment on a multi-domain server, refer to the system controller or system service processor documentation before beginning the installation process.

Using the Solaris Web Start Program to Install or Upgrade From a Remote DVD-ROM or CD-ROM

If you want to install the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive, you can use a drive that is attached to another machine. Both machines must be connected to the same subnet. Use the following instructions to complete the installation.

▼ To Install or Upgrade From a Remote DVD-ROM and CD-ROM

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

In the following procedure, the remote system with the DVD-ROM or CD-ROM is identified as *remote system*. The system that is the client to be installed is identified as *client system*.

1. **Identify a system that is running the Solaris operating environment and has a DVD-ROM or CD-ROM drive.**
2. **On the *remote system* with the DVD-ROM or CD-ROM drive, insert the Solaris 9 DVD or the Solaris 9 Installation CD in the drive.**

The Volume Manager mounts the disc.

3. **On the remote system, change directories to the DVD or CD where the `add_install_client` command is located.**

- For DVD media, type:

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

- For CD media, type:

```
remote system# cd /cdrom/cdrom0/s0
```

4. **On the remote system, add the system that you want to install as a client.**

- For DVD media, type:

```
remote system# ./add_install_client \
client_system_name arch
```

- For CD media, type:

```
remote system# ./add_install_client -s remote_system_name: \
/cdrom/cdrom0/s0 client_system_name arch
```

remote_system_name The name of the system with the DVD-ROM or CD-ROM drive

client_system_name The name of the machine you want to install

arch

The platform group of the machine you want to install, for example sun4u. On the system that you want to install, find the platform group by using the `uname -m` command.

5. Boot the *client system* that you want to install.

client system: `ok boot net`

The Solaris Web Start installation begins.

6. Follow the instructions to type system configuration information if needed.

- If you are using DVD media, follow the instructions on the screen to complete the installation. You are finished.
- If you are using CD media, the machine reboots and the Solaris installation program begins. After the Welcome panel, the Specify Media panel appears with Network File System selected. Proceed to Step 7.

7. On the Specify Media panel, click Next.

The Specify Network File System Path panel appears and the text field contains the installation path.

client_system_ip_address: `/cdrom/cdrom0/s0`

8. On the remote system where the DVD or CD is mounted, change directories to root.

remote system# `cd /`

9. On the remote system, check for the path to the slice that has been shared.

remote system# `share`

10. On the remote system, unshare the Solaris 9 DVD or Solaris 9 Installation CD by using the path that is found in Step 9. If paths lead to two slices, unshare both slices.

remote system# `unshare absolute_path`

absolute_path

Is the absolute path shown in the share command

In this example, slice 0 and slice 1 are unshared.

remote system# `unshare /cdrom/cdrom0/s0`

remote system# `unshare /cdrom/cdrom0/s1`

11. Eject the Solaris 9 Installation CD.

remote system# `eject cdrom`

12. On the remote system, insert the Solaris 9 Software 1 of 2 in the CD-ROM drive.

13. On the remote system, export the Solaris 9 Software 1 of 2 CD.

```
remote system# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
```

14. On the client system that you are installing, continue the Solaris installation by clicking Next.
15. If the Solaris Web Start program prompts you to insert the Solaris 9 Software 2 of 2 CD, repeat Step 9 through Step 14 to unshare the Solaris 9 Software 1 of 2 CD and to export and install the Solaris 9 Software 2 of 2 CD.
16. If the Solaris Web Start program prompts you to insert the Solaris 9 Languages CD, repeat Step 9 through Step 14 to unshare the Solaris 9 Software 2 of 2 CD and to export and install the Solaris 9 Languages CD.

When you export the Solaris 9 Languages CD, an installer window appears on the machine where the CD-ROM is mounted. Ignore the installer window while you install the Solaris 9 Languages CD. After you complete the installation of the Solaris 9 Languages CD, close the installer window.

Additional SvR4 Packaging Requirements (Reference)

This appendix is for system administrators who need to use the custom JumpStart program or Solaris Live Upgrade to install or remove packages, especially third-party packages. Following these packaging requirements keeps a custom JumpStart installation non-interactive and avoids modifying the currently running system so you can upgrade with Solaris Live Upgrade.

Note – An alternate root (/) is a copy of the operating environment, not the currently running system.

Packaging Requirements Overview

For the custom JumpStart program and Solaris Live Upgrade to work properly, packages must comply with the SvR4 packaging requirements. The *Application Packaging Developer's Guide* provides more specific information on packaging requirements and definitions of terminology. See especially the chapter: "Advanced Package Creation Techniques" in *Application Packaging Developer's Guide*.

For basic information on adding and removing packages and the installation administration file, see "Managing Software (Overview)" in the *System Administration Guide: Basic Administration*. Also see the specific man pages.

For detailed information on the commands referenced in this appendix, see the man pages, `dircmp(1)`, `fssnap(1M)`, `ps(1)`, or `truss(1)`.

Table C-1 lists information that applies to either Solaris Live Upgrade or the custom JumpStart program.

TABLE C-1 Requirements Information

Installation Method	Requirements Documented
Solaris Live Upgrade	<ul style="list-style-type: none">■ See SvR4 packaging requirements in the <i>Application Packaging Developer's Guide</i>.■ See "Custom JumpStart Program and Solaris Live Upgrade Alternate Root (/) Requirements" on page 352.■ See "Solaris Live Upgrade Alternate Boot Environment Compliance" on page 354.
Custom JumpStart program	<ul style="list-style-type: none">■ See SvR4 packaging requirements in the <i>Application Packaging Developer's Guide</i>.■ See "Custom JumpStart Program and Solaris Live Upgrade Alternate Root (/) Requirements" on page 352.■ See "Upgrade Compliance With the Custom JumpStart Program" on page 356.

Custom JumpStart Program and Solaris Live Upgrade Alternate Root (/) Requirements

An alternate root (/) is a copy of the operating environment, not the currently running system. A package to be used by Live Upgrade or the custom JumpStart program must follow these requirements:

- Enable a custom JumpStart installation or upgrade with no user interaction
- Make no modification of the current running system, which is necessary when using Solaris Live Upgrade

The following list explains the requirements for alternate root (/) compliance.

- For an installation of an operating system to be successful, packages must recognize and correctly respect alternate root (/) specifiers.
Packages can include absolute paths in their `pkgmap` file (package map). If these files exist, they are written relative to the `-R` option of the `pkgadd` command. Packages that contain both absolute and relative (relocatable) paths can be installed to an alternative root (/) as well. `$PKG_INSTALL_ROOT` is prepended to both absolute and relocatable files so all paths are resolved properly when being installed by `pkgadd`.
- Packages being installed using the `pkgadd -R` option or being removed using the `pkgrm -R` option must not alter the currently running system.

Any procedure scripts that come with the packages being installed with the `pkgadd` command `-R` option or being removed using the `pkgrm` command `-R` option must not alter the currently running system. Any installation scripts that you provide must reference any directory or file prefixed with the `$PKG_INSTALL_ROOT` variable. The package must write all directories and files with the `$PKG_INSTALL_ROOT` prefix. The package must not remove directories without a `$PKG_INSTALL_ROOT` prefix. Table C-2 provides examples of correct scripting syntax.

TABLE C-2 Examples of Installation Script Syntax

Script Type	Correct Syntax	Incorrect Syntax
Bourne shell "if" statement fragments	<pre>if [-f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf] ; then</pre>	<pre>if [-f /etc/myproduct.conf]\ ; \ then</pre>
Removing a file	<pre>/bin/rm -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>/bin/rm -f /etc/myproduct.conf</pre>
Changing a file	<pre>echo "test=no" \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>echo "test=no"\ /etc/myproduct.conf</pre>

Differences Between `$PKG_INSTALL_ROOT` and `$BASEDIR` Overview

`$PKG_INSTALL_ROOT` is the location of the root (`/`) file system of the machine to which you are adding the package. The location is set to the `-R` argument to the `pkgadd` command. For example, if the following command is invoked, then `$PKG_INSTALL_ROOT` is prepended to `/a` during the installation of the package.

```
# pkgadd -R /a SUNWvxvm
```

`$BASEDIR` points to the *relocatable* base directory into which relocatable package objects are installed. Only relocatable objects are installed here. Non-relocatable objects (those that have *absolute* paths in the `pkgmap` file) are always installed relative to the alternate root (`/`), but not relative to the `$BASEDIR` in effect. If a package has no relocatable objects, then the package is said to be an absolute package (or non-relocatable), and `$BASEDIR` is undefined and not available to package procedure scripts.

For example, suppose a package's `pkgmap` file has two entries:

```
1 f none sbin/ls 0555 root sys 3541 12322 1002918510
1 f none /sbin/ls2 0555 root sys 3541 12322 2342423332
```

The `pkginfo` file has a specification for `$BASEDIR`:

```
BASEDIR=/opt
```

If this package is installed with the following command, then `ls` is installed in `/a/opt/sbin/ls`, but `ls2` is installed as `/a/sbin/ls2`.

```
# pkgadd -R /a SUNWtest
```

Solaris Live Upgrade Alternate Boot Environment Compliance

When using Solaris Live Upgrade and creating a new boot environment, avoid problems by following these guidelines.

- Your package procedure scripts must be independent of the current active operating environment. Procedure scripts define actions that occur at particular points during package installation and removal. Four procedure scripts can be created with these predefined names: `preinstall`, `postinstall`, `preremove`, and `postremove`. Package procedure scripts must be independent of the currently active operating environment because an alternate boot environment could be switched by using Solaris Live Upgrade.
 - These scripts must not start or stop any processes or depend on the output of commands such as `ps` or `truss`, which are operating-system dependent and report information about the currently running system.
 - Procedure scripts are free to use other standard UNIX commands such as `expr`, `cp`, and `ls` and other commands that facilitate shell scripting. But, the current alternate root (`/`) must not be modified except within the rules outlined in the section, “Custom JumpStart Program and Solaris Live Upgrade Alternate Root (`/`) Requirements” on page 352.
 - All scripts must be written in Bourne shell (`/bin/sh`). Bourne shell is the interpreter used by the `pkgadd` command to execute the procedure scripts.
 - Package procedure scripts must *not* invoke commands that do not exist in releases previous to the 2.6 release. For example, package procedure scripts cannot invoke the `pgrep` command. Since the 2.6 release, many commands have had additional features added. Package procedure scripts must not use any command options that did not exist in the 2.6 release. For example, the `-f` option is new to the `umount` command.
- 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
```

<i>dir_name</i>	Specifies the name of the directory where the package resides
<i>pkg_name</i>	Specifies the name of the package

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

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

No errors should be displayed.

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

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

No errors should be displayed.

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

These requirements for creating, modifying, and deleting files can be verified using a variety of commands. For example, the `dircmp` or `fssnap` commands can be used to verify that packages behave properly. Also, the `ps` command can be used for testing daemon compliance by making sure daemons are not stopped or started by the package. The `truss` and `pkgadd` 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-`BASEDIR` access and shows only non-read-only access to paths that do not lie within the specified alternate root (`/`).

```
# BASEDIR=/a; export BASEDIR
# truss -t open /usr/sbin/pkgadd -R ${BASEDIR} SUNWvxvm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v \
'open("${BASEDIR}
```

For detailed information on the commands referenced in this section, see the man pages, `dircmp(1)`, `fssnap(1M)`, `ps(1)`, or `truss(1)`.

Upgrade Compliance With the Custom JumpStart Program

Custom JumpStart program compliance ensures that packages can be added and removed while being part of the traditional Solaris installation utilities, which are the following:

- The custom JumpStart program
- Solaris `suninstall` program
- Solaris Web Start installation method

The custom JumpStart program compliance also ensures that the package can participate in Solaris upgrades. To be custom JumpStart program-compliant, a package also must follow alternate root (/) requirements that are outlined in “Custom JumpStart Program and Solaris Live Upgrade Alternate Root (/) Requirements” on page 352.

To be able to use the custom JumpStart program effectively, packages must be added or removed without the user being prompted for information. To prevent user interaction, set up a new administration file with the `pkgadd` command `-a` option. The `-a` option defines an installation administration file to be used in place of the default administration file. Using the default file might result in the user being prompted for more information. You can create an administration file that indicates to `pkgadd` that it should bypass these checks and install the package with user confirmation. The following examples show how to use the `pkgadd` administration file.

- If no administration file is provided, `pkgadd` uses `/var/sadm/install/admin/default`. Using this file could 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/nocheck`.

```
# pkgadd -a /tmp/nocheck
```

The following is an example of an installation administration file that prevents `pkgadd` from prompting the user for confirmation before installing the package.

```
mail=  
instance=overwrite  
partial=nocheck  
runlevel=nocheck  
idepend=nocheck  
space=nocheck  
setuid=nocheck  
conflict=nocheck  
action=nocheck  
basedir=default
```

For details, see the man pages, `admin(4)` or `pkgadd(1M)`.

Upgrading to a Solaris Update Release (Tasks)

This chapter provides instructions on checking patches before upgrading to an update to the Solaris operating environment.

Upgrading to a Solaris Update Release

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

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

The Patch Analyzer performs an analysis on your system to determine which patches, if any, will be removed by upgrading to the Solaris 9 Update release. The Patch Analyzer is available in the following formats.

- If you are using the Solaris Web Start program to upgrade, the Patch Analyzer dialog box appears. Select Yes to perform the analysis.
- If you are using the Solaris `suninstall` program to upgrade, select Analyze on the Patch Analysis dialog box to perform the analysis.
- If you are using a custom JumpStart installation or Solaris Live Upgrade to upgrade, run the `analyze_patches` script to perform the analysis. For detailed instructions, see “To Run the `analyze_patches` Script” on page 360.

After you perform the analysis, refer to “To Review the Patch Analyzer Output” on page 360 for detailed information about the patch analysis results.

▼ To Run the `analyze_patches` Script

Note – To run the `analyze_patches` script, the installed system and the Solaris 9 DVD, Solaris 9 Software CDs, or net image must be accessible by the script either through NFS or locally mounted media.

1. Change to the `Misc` directory.

- If the image is located on locally mounted media, type:

```
# cd /cdrom/sol_9_Update_sparc/s9/Solaris_9/Misc
```

In this command, *Update* is the actual Update identifier.

- If the image is available on an NFS file system, type:

```
# cd /NFS_mount_directory/Solaris_9/Misc
```

2. Run the `analyze_patches` script:

```
# ./analyze_patches [-R rootdir] [-N netdir] [-D databasedir]
```

<code>-R rootdir</code>	<code>rootdir</code> is the root of the installed system. The default is <code>/</code> .
<code>-N netdir</code>	<code>netdir</code> is the path to the root of the OS image to be installed. The default is <code>/cdrom/cdrom0</code> . <code>netdir</code> is the path to the directory that contains the <code>solaris_9</code> directory. You must use this option if you are running the <code>patch_analyzer</code> from an NFS mount point.
<code>-D databasedir</code>	If the script is invoked from a directory other than the <code>Misc/</code> directory in the OS image, the program cannot find the database it uses for patch analysis. Use the <code>-D</code> option to supply the path to the database. Without this database, which is located in <code>Solaris_9/Misc/database</code> on the OS image, the script does not work properly.

▼ To Review the Patch Analyzer Output

After you perform the analysis, use these steps to review the output.

1. Review the output of the Patch Analyzer.

The Patch Analyzer provides a list of patches that will be removed, downgraded, accumulated, or obsoleted by other patches. Patch accumulations are similar to patch upgrades. The accumulated patch is removed and its fixes are delivered by a new patch. Messages such as the following are shown:

```
Patch 105644-03 will be removed.  
Patch 105925 will be downgraded from -02 to -01.  
Patch 105776-01 will be accumulated/obsoleted by patch 105181-05.
```


If the Patch Analyzer program does not provide a list, no action is taken against any patches that were previously installed on your system.

2. Decide if the patch replacements and deletions are acceptable.

- If yes, upgrade the system.
- If no, do not upgrade the system.

At an update release, instead of upgrading, you can use the Solaris 9 Maintenance Update release to apply only patches to your system.

Note – The Solaris 9 Maintenance Update is available on CD and by download. Instructions for applying patches are provided in the *Maintenance Update Installation Guide*.

Solaris 9 Updates

This appendix describes changes to this book for Solaris 9 update releases.

Solaris 9 9/02 release.

Minor changes were made to fix a bug.

Glossary

active boot environment	The boot environment that is currently booted.
alternate root (/) file system	A copy of the operating environment, not the currently running system.
archive	A file that contains all of the files that were copied from a master system. The file also contains identification information about the archive, such as name and the date you created the archive. When you select to install an archive on a system, the system then contains the exact configuration of the master system you used to create the archive.
arrow keys	One of the four directional keys on the numeric keypad.
begin script	A user-defined Bourne shell script, specified within the <code>rules</code> file, that performs tasks before the Solaris software is installed on the system. You can use begin scripts only with custom JumpStart installations.
boot	To load the system software into memory and start it.
boot environment	A bootable Solaris environment that consists of a set of disk slices and the appropriate mount points and file systems. These disk slices might be on the same disk or distributed across multiple disks.
boot server	A server system that provides client systems on the same network subnet with the programs and information that they need to start. A boot server is required to install over the network if the install server is on a different subnet than the systems on which Solaris software is to be installed.
checksum	The result of adding a group of data items that are used for checking the group. The data items can be either numerals or other character strings that are treated as numerals during the checksum calculation. The checksum value verifies that communication between two devices is successful.

client	In the client-server model for communications, the client is a process that remotely accesses resources of a compute server, such as compute power and large memory capacity.
clone system	A system that you installed by using a Web Start Flash archive. The clone system has the same installation configuration as the master system.
cluster	A logical collection of packages (software modules). The Solaris 9 software is divided into <i>software groups</i> , which are each composed of clusters and <i>packages</i> .
command line	A string of characters that begins with a command, often followed by arguments, including options, file names, and other expressions, and terminated by the end-of-line character.
Core	A software group that contains the minimum software that is required to boot and run the Solaris operating environment on a system. Core includes some networking software and the drivers that are required to run the Common Desktop Environment (CDE) desktop. Core does not include the CDE software.
critical file systems	File systems that are required by the Solaris operating environment. When you use Solaris Live Upgrade, these file systems are separate mount points in the <code>vfstab</code> of the active and inactive boot environments. Examples are <code>root (/)</code> , <code>/usr</code> , <code>/var</code> , or <code>/opt</code> . These file systems are always copied from the source to the inactive boot environment.
custom JumpStart	A type of installation in which the Solaris 9 software is automatically installed on a system based on a user-defined profile. You can create customized profiles for different types of users and systems. A custom JumpStart installation is a JumpStart installation you create.
custom probes file	A file, which must be located in the same JumpStart directory as the <code>rules</code> file, is a Bourne shell script that contains two types of functions: probe and comparison. Probe functions gather the information you want or do the actual work and set a corresponding <code>SI_</code> environment variable you define. Probe functions become probe keywords. Comparison functions call a corresponding probe function, compare the output of the probe function, and return 0 if the keyword matches or 1 if the keyword doesn't match. Comparison functions become rule keywords. See also <i>rules file</i> .
derived profile	A profile that is dynamically created by a begin script during a custom JumpStart installation.
Developer Solaris Software Group	A software group that contains the End User Solaris Software Group plus the libraries, include files, man pages, and programming tools for developing software.

DHCP	Dynamic Host Configuration Protocol is an application-layer protocol that enables individual computers, or clients, on a TCP/IP network to extract an IP address and other network configuration information from a designated and centrally maintained DHCP server or servers. This facility reduces the overhead of maintaining and administering a large IP network.
disc	An optical disc, as opposed to a magnetic disk, which recognizes the common spelling that is used in the compact disc (CD) market. For example, a CD-ROM is an optical disc.
disk	A round platter, or set of platters, of a magnetized medium that is organized into concentric tracks and sectors for storing data such as files. See also disc.
disk configuration file	A file that represents a structure of a disk (for example, bytes/sector, flags, slices). Disk configuration files enable you to use <code>pfinstall</code> from a single system to test profiles on different-size disks.
diskless client	A client on a network that relies on a server for all of its disk storage.
domain	A part of the Internet naming hierarchy. A domain represents a group of systems on a local network that share administrative files.
domain name	The name that is assigned to a group of systems on a local network that share administrative files. The domain name is required for the Network Information Service (NIS) database to work properly. A domain name consists of a sequence of component names that are separated by periods (for example: <code>tundra.mpk.ca.us</code>). As you read a domain name from left to right, the component names identify more general (and usually remote) areas of administrative authority.
End User Solaris Software Group	A software group that contains the Core software group plus the recommended software for an end user, including the Common Desktop Environment (CDE) and DeskSet software.
Entire Solaris Software Group	A software group that contains the entire Solaris 9 release.
Entire Solaris Software Group Plus OEM Support	A software group that contains the entire Solaris 9 release, plus additional hardware support for OEMs. This software group is recommended when installing Solaris software on SPARC based servers.
<code>/etc</code>	A directory that contains critical system configuration files and maintenance commands.
<code>/export</code>	A file system on an OS server that is shared with other systems on a network. For example, the <code>/export</code> file system can contain the root file system and swap space for diskless clients and the home directories for users on the network. Diskless clients rely on the <code>/export</code> file system on an OS server to boot and run.

fallback	A reversion to the environment that ran previously. Use fallback when you are activating an environment and the boot environment that is designated for booting fails (or shows some undesirable behavior).
file server	A server that provides the software and file storage for systems on a network.
file system	In the SunOS™ operating system, a tree-structured network of files and directories that you can access.
finish script	A user-defined Bourne shell script, specified within the <code>rules</code> file, that performs tasks after the Solaris software is installed on the system, but before the system reboots. You can use finish scripts only with custom JumpStart installations.
format	To put data into a structure or divide a disk into sectors for receiving data.
function key	One of the 10 or more keyboard keys that are labeled F1, F2, F3, and so on that are mapped to particular tasks.
hard link	A directory entry that references a file on disk. More than one such directory entry can reference the same physical file.
host name	The name by which a system is known to other systems on a network. This name must be unique among all the systems within a particular domain (usually, this means within any single organization). A host name can be any combination of letters, numbers, and minus signs (-), but it cannot begin or end with a minus sign.
HTTP	(Hypertext Transfer Protocol) (n.) The Internet protocol that fetches hypertext objects from remote hosts. This protocol is based on TCP/IP.
inactive boot environment	Any boot environment that is not currently booted and not designated for activation on next reboot.
initial installation option	An option that is presented by the Solaris Web Start program and the Solaris <code>suninstall</code> program that overwrites the disk(s) with a new version of Solaris. The initial installation option is presented for systems that can be upgraded. However, the disk(s) that contain the old version of Solaris software (including the local modifications) are overwritten if you choose the initial installation option.
install server	A server that provides the Solaris 9 CD images from which other systems on a network can install Solaris (also known as a <i>media server</i>). You can create an install server by copying the Solaris 9 CD images to the server's hard disk.
IP address	Internet protocol address. In TCP/IP, a unique 32-bit number that identifies each host in a network. An IP address consists of four numbers that are separated by periods (192.9.9.1, for example). Most

often, each part of the IP address is a number between 0 and 225. However, the first number must be less than 224 and the last number cannot be 0.

IP addresses are logically divided into two parts: the network (similar to a telephone area code), and the local system on the network (similar to a phone number). The numbers in a Class A IP address, for example, represent "network.local.local.local" and the numbers in a Class C IP address represent "network.network.network.local."

Class	Range (<i>xxx</i> is a number 0 to 255)	Number of Available IP Addresses
Class A	1. <i>xxx.xxx.xxx</i> - 126. <i>xxx.xxx.xxx</i>	Over 16 million
Class B	128.0. <i>xxx.xxx</i> - 191.255. <i>xxx.xxx</i>	Over 65,000
Class C	192.0.0. <i>xxx</i> - 223.255.255. <i>xxx</i>	256

- IPv6** IPv6 is a new version (version 6) of Internet Protocol (IP) that is designed to be an evolutionary step from the current version, IPv4 (version 4). IPv6 is an increment to IPv4. Deploying IPv6, using defined transition mechanisms, does not disrupt current operations. In addition, IPv6 provides a platform for new Internet functionality.
- IPv6 is described in more detail in "Overview of IPv6" in *System Administration Guide, Volume 3*.
- job** A user-defined task to be completed by a computer system.
- JumpStart directory** When you use a profile diskette for custom JumpStart installations, the JumpStart directory is the root directory on the diskette that contains all the essential custom JumpStart files. When you use a profile server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files.
- JumpStart installation** A type of installation in which the Solaris 9 software is automatically installed on a system by using the factory-installed JumpStart software.
- Kerberos** A network authentication protocol that uses strong, secret-key cryptography to enable a client and server to identify themselves to each other over an insecure network connection.
- locale** A geographic or political region or community that shares the same language, customs, or cultural conventions (English for the U.S. is en_US, and English for the U.K. is en_UK).

master system	A system that you use to create a Web Start Flash archive. The system configuration is saved in the archive.
media server	See <i>install server</i> .
metadevice	See <i>volume</i> .
miniroot	The smallest possible bootable Solaris <code>root</code> file system. A miniroot contains a kernel and just enough software to install the Solaris environment on a hard disk. The miniroot is the file system that is copied to a machine in the initial installation.
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.
name server	A server that provides a name service to systems on a network.
name service	A distributed network database that contains key system information about all the systems on a network so that the systems can communicate with each other. With a name service, the system information can be maintained, managed, and accessed on a network-wide basis. Sun supports the following name services: NIS and NIS+. Without a name service, each system has to maintain its own copy of the system information (in the local <code>/etc</code> files).
networked systems	A group of systems (called hosts) that are connected through hardware and software, so that they can communicate and share information. Referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.
network installation	A way to install software over the network—from a system with a CD-ROM drive to a system without a CD-ROM drive. Network installations require a <i>name server</i> and an <i>install server</i> .
NIS	The SunOS™ 4.0 (minimum) Network Information Service. A distributed network database that contains key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.
NIS+	The SunOS 5.0 (minimum) Network Information Service. NIS+ replaces NIS, the SunOS 4.0 (minimum) Network Information Service.
non-networked systems	Systems that are not connected to a network or do not rely on other systems.
<code>/opt</code>	A file system that contains the mount points for third-party and unbundled software.

OS server	A system that provides services to systems on a network. To serve diskless clients, an OS server must have disk space set aside for each diskless client's root file system and swap space (<code>/export/root</code> , <code>/export/swap</code>).
package	A collection of software that is grouped into a single entity for modular installation. The Solaris 9 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 Solaris <code>suninstall</code> program or the Solaris Web Start program. The patch analyzer performs an analysis on your system to determine which (if any) patches will be removed by upgrading to a Solaris 9 Update.
platform group	A vendor-defined grouping of hardware platforms for the purpose of distributing specific software. Examples of valid platform groups are <code>i86pc</code> and <code>sun4u</code> .
platform name	The output of the <code>uname -i</code> command. For example, the platform name for the Ultra 60 is <code>SUNW,Ultra-60</code> .
Power Management	Software that automatically saves the state of a system and turns it off after it is idle for 30 minutes. When you install the Solaris software on a system that complies with Version 2 of the U.S. Environmental Protection Agency's Energy Star guidelines—a <code>sun4u</code> SPARC system, for example—the Power Management software is installed by default. After a subsequent reboot, you are prompted to enable or disable the Power Management software.
	Energy Star guidelines require that systems or monitors automatically enter a "sleep state" (consume 30 watts or less) after the system or monitor becomes inactive.
probe keyword	A syntactical element that extracts attribute information about a system without the need to set up a matching condition and run a profile as required for a rule. See also <i>rule</i> .
profile	A text file that defines how to install the Solaris software (for example, which software group to install). Every rule specifies a profile that defines how a system is to be installed when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule. See also <i>rules file</i> .
profile diskette	A diskette that contains all the essential custom JumpStart files in its root directory (JumpStart directory).
profile server	A server that contains all the essential custom JumpStart files in a JumpStart directory.
/ (root)	In a hierarchy of items, the one item from which all other items are descended. The root item has nothing above it in the hierarchy. <code>/</code> is the base directory from which all other directories stem, directly or

indirectly. The root directory contains the directories and files critical for system operation, such as the kernel, device drivers, and the programs that are used to start (boot) a system.

rule	A series of values that assigns one or more system attributes to a profile.
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 <code>rules</code> file links each group to a profile, which is a text file that defines how the Solaris 9 software is to be installed on each system in the group. See also <i>profile</i> .
rules.ok file	A generated version of the <code>rules</code> file. The <code>rules.ok</code> file is required by the custom JumpStart installation software to match a system to a profile. You <i>must</i> use the <code>check</code> script to create the <code>rules.ok</code> file.
server	A network device that manages resources and supplies services to a client.
shareable file systems	File systems that are user-defined files such as <code>/export/home</code> and <code>/swap</code> . These file systems are shared between the active and inactive boot environment when you use Solaris Live Upgrade. Shareable file systems contain the same mount point in the <code>vfstab</code> in both the active and inactive boot environments. Updating shared files in the active boot environment also updates data in the inactive boot environment. Shareable file systems are shared by default, but you can specify a destination slice, and then the file systems are copied.
slice	The unit into which the disk space is divided by the software.
software group	A logical grouping of the Solaris software (clusters and packages). During a Solaris installation, you can install one of the following software groups: Core, End User Solaris Software, Developer Solaris Software, or Entire Solaris Software, and Entire Solaris Software Group Plus OEM Support.
Solaris 9 CD images	The Solaris 9 software that is installed on a system, which you can access on the Solaris 9 CDs or an install server's hard disk to which you have copied the Solaris 9 CD images.
Solaris <code>suninstall</code> program	A command-line interface (CLI), menu-driven, interactive script that enables you to set up a system and install the Solaris 9 software on it.
Solaris Web Start program	A graphical user interface (GUI) or command-line interface (CLI) installation program that uses wizard panels to guide you step-by-step through installing the Solaris 9 software and third-party software.
standalone	A computer that does not require support from any other machine.
subnet	A working scheme that divides a single logical network into smaller physical networks to simplify routing.

subnet mask	A bit mask that is used to select bits from an Internet address for subnet addressing. The mask is 32 bits long and selects the network portion of the Internet address and 1 or more bits of the local portion.
swap space	A slice or file that temporarily holds the contents of a memory area till it can be loaded back into memory. Also known as the <code>/swap</code> or <code>swap</code> file system.
sysidcfg file	A file in which you specify a set of special system configuration keywords that preconfigure a system.
time zone	Any of the 24 longitudinal divisions of the earth's surface for which a standard time is kept.
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.
upgrade option	An option that is presented by the Solaris Web Start program and the Solaris <code>suninstall</code> program. The upgrade procedure merges the new version of Solaris with existing files on your disk(s). An upgrade also saves as many local modifications as possible since the last time Solaris was installed.
<code>/usr</code>	A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large <code>/usr</code> file system with a server rather than maintaining a local copy minimizes the overall disk space that is required to install and run the Solaris 9 software on a system.
utility	A standard program, usually furnished at no charge with the purchase of a computer, that does the computer's housekeeping.
<code>/var</code>	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, <code>vi</code> files, mail files, and <code>uucp</code> files.
volume	A group of physical slices that are accessed as a single logical device by concatenation, striping, mirroring, setup RAID5 volumes, or logging physical devices. After they are created, volumes are used like slices. The volume maps logical block address to the correct location on one of the physical devices. The type of mapping depends on the configuration of the particular volume. Also known as pseudo device or virtual device in standard UNIX terms.
Volume Manager	A program that provides a mechanism to administer and obtain access to the data on CD-ROMs and diskettes.
Web Start Flash	A Solaris installation feature that enables you to create an archive of the files on a system, known as the master system. You can then use

the archive to install other systems, making the other systems identical in their configuration to the master system.

Index

Numbers and Symbols

- #
 - in profiles, 151
 - in rules files, 147
- = (equal sign) in profile field, 160
- ! (exclamation mark) rule field, 148
- && (ampersands) rule field, 148
- (/) file systems
 - value set by JumpStart, 215

A

- activating a boot environment for Live
 - Upgrade, overview, 233
- activation failure of boot environment for Live
 - Upgrade, overview, 235
- add_install_client command
 - description, 91
 - example
 - boot server for DVD media, 77
 - for boot server for CD media, 89
 - same subnet for CD media, 89
 - with DHCP for CD media, 89
 - with DHCP for DVD media, 77
 - install server setup
 - with CD media, 88
 - with DVD media, 76
 - JumpStart directory access, 143
- add_to_install_server, description, 91
- adding
 - clusters when upgrading, 205
 - adding (Continued)
 - dataless clients
 - with CD media, 87
 - with DVD media, 75
 - files with a finish script, 161
 - install server configuration
 - with CD media, 88
 - with DVD media, 76
 - locale.org_dir table entries, 56
 - packages and patches with a finish script, 162
 - packages from software groups, 213
 - rules to rules file, 148
 - systems from network, 70, 80
- alternative installation programs, 169
- ampersands (&&) rule field, 148
- AND rule field, 148
- any
 - probe keyword
 - description and values, 221
 - rule keyword
 - description and values, 189, 220
- arch probe keyword, 220
- arch rule keyword, 189, 220
- archive
 - See Web Start Flash
- archive_location keyword, 196
- auto_install_sample directory
 - check script, 158, 175
 - copying files to JumpStart directory, 142, 146
 - set_root_pw finish script, 164, 165

B

- b option of `setup_install_server` command, 184
- backslash in `rules` files, 147
- `backup_media` keyword, 61, 201
- `banner` command, 92
- `begin.log` file, 159
- `begin` rule field, description, 148
- `begin` scripts
 - creating derived profiles with, 160
 - overview, 159
 - permissions, 159
 - rule field, 148
 - site-specific installation programs, 169
- `boot: cannot open /kernel/unix` message, 338
- `boot_device` keyword, 202
- boot environment, view status, 305
- boot failure of boot environment for Live Upgrade, overview, 235
- boot server
 - creating on subnet
 - with DVD media, 73
 - creating on subnet with CD media, 85
 - creating with DVD, example, 75
 - description, 66
 - requirement for network installation, 66
- booting the system, resetting terminals and display first, 92
- `bootparams` file
 - enabling JumpStart directory access, 144
 - updating, 341
- Bourne shell scripts in rule fields, 148

C

- c option
 - `pfinstall` command, 156
 - `add_install_client` command, 88, 187, 188
- Can't boot from file/device message, 338
- cancel a Solaris Live Upgrade job, 296
- cannot open /kernel/unix message, 338
- CHANGE DEFAULT BOOT DEVICE message, 342
- change name of Solaris Live Upgrade boot environment, 301

- changing directories
 - image of Solaris 9 software on local disk, 146
 - to image of Solaris 9 software on local disk, 142
 - to JumpStart directory, 158, 175
- check script
 - `custom_probes` file validation, 175, 176
 - `custom_probes.ok` file creation, 175
 - derived profiles and, 160
 - `rules` file validation, 176
 - `rules` file validation, 157, 158
 - `rules.ok` file creation, 157
 - testing rules, 158, 176
- `client_arch` keyword, 203
- `client_name`, description, 89
- `client_root` profile keyword, 203
- clock gained xxx days message, 338
- cluster profile keyword
 - description and values, 204, 205
 - examples, 151
- commands for Solaris Live Upgrade, 315
- comments
 - in profiles, 151
 - in `rules` files, 147
- compare Live Upgrade boot environments, 297
- configuring
 - creating disk configuration files, 167
 - Solaris Live Upgrade file systems, 253
- copying
 - JumpStart directory files, 162
 - JumpStart installation files, 142, 146
 - Solaris Live Upgrade file systems, 294
- Core Solaris Software Group, 204
- CPUs (processors)
 - probe keywords, 220
 - rule keywords, 189, 220
- creating
 - a boot environment for Live Upgrade, 226, 251
 - boot server on a subnet
 - tasks, CD media, 80
 - tasks, DVD media, 70
 - boot server on subnet
 - with CD media, 85
 - boot server on subnet with DVD media, 73
 - /etc/locale file, 55

- creating (Continued)
 - custom_probes.ok file, 175
 - disk configuration files, 167
 - install server with CD media, 80, 81
 - install server with DVD media, 69, 71
 - JumpStart directory
 - server, 141
 - local file systems, 207
 - profiles
 - derived, 160
 - description, 150
 - rules file, 147
 - rules.ok file, 175
 - rules.ok file, 157
 - sysidcfg file, 53
 - UFS, 145
 - critical file systems, definition, 226
 - .cshrc file, 163
 - custom JumpStart installation, 177
 - booting and installing, 177
 - compared to other installation methods, 28
 - description, 138
 - examples, 181
 - booting and installing, 188
 - check script, 187
 - eng_profile creation, 185
 - engineering systems setup, 187
 - JumpStart directory, 184, 185
 - marketing_profile creation, 185
 - marketing systems setup, 184
 - networked, 137
 - non-networked, 134
 - rules file editing, 186
 - site setup, 181, 182
 - standalone system, 134
 - installing Web Start Flash archives, 120
 - optional features, 159
 - begin scripts, 159, 160
 - finish scripts, 161, 165
 - overview, 159
 - site-specific installation programs, 169
 - overview, 137
 - preparing, 138, 158
 - tip line connection and, 179
 - when upgrading, 60
 - custom_probes file
 - naming, 172
 - requirements, 172

- custom_probes file (Continued)
 - testing custom_probes, 176
 - validating by using check, 175, 176
- custom_probes.ok file
 - creating, 175
 - description, 175

D

- d option, add_install_client command, 88
- daemons, 341
- date and time, preconfiguring, 48
- default router, preconfiguring, 48
- defaults
 - derived profile name, 160
 - partitioning, 214
 - designating disks, 217
 - excluding disks, 206
 - SI_CONFIG_DIR variable, 161
 - software group installed, 205
- delete a Live Upgrade boot environment, 299
- deleting
 - clusters when upgrading, 205
 - packages from software groups, 213
- derived profiles, 160
- Developer Solaris Software Group, 204
 - description, 34
 - profile example, 151
 - size, 35
- dfstab file, 142, 185
- DHCP (Dynamic Host Configuration Protocol),
 - preconfiguring, 48
- directories
 - changing
 - to image of Solaris 9 software on local disk, 142, 146
 - to JumpStart directory, 158, 175
 - JumpStart
 - adding files, 161, 162
 - allowing access, 143
 - copying files, 162
 - copying installation files, 142, 146
 - creating directory, 184
 - creating for systems, 144
 - permissions, 141, 145
 - rules file example, 147
 - sharing directory, 142, 185

- disk configuration files
 - creating, 167
 - description, 154, 167
- disk space
 - planning, 33
 - requirements for software groups, 35
- diskettes
 - formatting, 145
 - JumpStart directory access, 143
- diskless clients
 - platforms, 203
 - swap space, 204
- disks probe keyword, description and values, 220
- disksize rule keyword
 - description and values, 190, 220
- display, tip line connection and custom
 - JumpStart installation, 179
- display name of Live Upgrade boot environment, 300
- displaying
 - mounted file systems, 91
 - platform name, 92
 - system information, 92
- domain name, preconfiguring, 48
- domainname probe keyword, 220
- domainname rule keyword, 190, 220
- domains
 - probe keyword, 220
 - rule keyword, 190, 220
- dontuse profile keyword, 206, 217

E

- End User Solaris Software Group, 204
 - description, 34
- End User Solaris Software Group, size, 35
- eng_profile example, 185
- Entire Solaris Software Group, 204
 - description, 34
- Entire Solaris software group, size, 35
- Entire Solaris Software Group Plus OEM Support, 204
 - description, 34
 - size, 35
- equal sign (=) in profile field, 160

- /etc/bootparams file
 - enabling JumpStart directory access, 144, 341
- /etc/dfs/dfstab file, 142, 185
- /etc/locale file, 55
- /etc/mnttab file, 145
- exclamation mark (!) rule field, 148
- existing partitioning value, 214
- explicit partitioning value, 215
- /export file system, 33

F

- factory JumpStart, compared to other
 - installation methods, 30
- failed upgrade
 - rebooting problems, 343, 344
 - recovery with Solaris Live Upgrade, 286
- fdformat command, 145
- files and file systems
 - begin script output, 159
 - copying
 - a shareable file system for Solaris Live Upgrade, 266
 - JumpStart directory files using finish scripts, 162
 - JumpStart installation files, 142, 146
 - creating local file systems, 207
 - customizing for Live Upgrade, 253
 - displaying mounted file systems, 91
 - finish script output, 161
 - mounting remote file systems, 206
 - overview for Live Upgrade, 226, 227
 - sharing file systems in Live Upgrade,
 - overview, 245
 - UFS creation, 145
- filesys keyword, 207
- filesys profile keyword
 - description and values, 206
 - examples, 151
- finish.log file, 161
- finish rule field, description, 149
- finish scripts
 - adding files, 161
 - adding packages and patches, 162
 - customizing the root environment, 163
 - rule field, 149

finish scripts (Continued)
 setting the system's root password, 164
flar command, 128
flarcreate command, 126
Flash
 See Web Start Flash
formatting diskettes, 145

G

geo keyword, 209
getfile: RPC failed: error 5: RPC Timed out
 message, 144

H

hard disks
 mounting, 206
 partitioning
 designating for partitioning default, 217
 examples, 151
 excluding for partitioning default, 206
 profile keyword, 214
 rootdisk values, 215
 size
 probe keywords, 220, 221
 root space, 203
 rule keywords, 190, 194, 220, 221
 space available, 71, 82, 86
 swap space
 diskless client, 204
 maximum size, 204
 profile examples, 138, 151
host name, preconfiguring, 48
hostaddress probe keyword, 220
hostaddress rule keyword, 190, 220
hostname probe keyword, description and
 values, 220
hostname rule keyword
 description and values, 191, 220
 example, 189

I

install_config command, 144

install server
 creating with CD media, 81
 creating with CD media, example, 84
 creating with DVD media, 71
 creating with DVD media, example, 73
 network installation setup with CD media,
 88
 network installation setup with DVD media,
 76
 on subnet, 73, 84
 system types applicable, 65
install_type keyword, 210
install_type profile keyword
 examples, 151
 requirement, 150, 151
 testing profiles, 155, 156
installation
 compared to upgrade, 23
 comparison of methods, 27
 disk space recommendations, 33
 overview of tasks, 21
 system requirements, 24
installed probe keyword, description and
 values, 220
installed rule keyword
 description and values, 191, 220
IP addresses
 preconfiguring, 48
 probe keyword, 220
 rule keyword, 190, 220
IPv6, preconfiguring, 48
isa_bits keyword, 211

J

JumpStart directory
 adding files with finish scripts, 161, 162
 copying files
 installation files, 142, 146
 using finish scripts, 162
 creating
 diskette, 145
 example, 184
 server, 141
 permissions, 141, 145
 rules file example, 147
 sharing, 141, 185

K

- karch probe keyword, 220
- karch rule keyword, 191, 220
- Kerberos
 - information to configure, 39
 - preconfiguring, 48
- keywords
 - custom JumpStart
 - Web Start Flash archives, 196
 - probe, 171
 - sysidcfg file, 50

L

- layout_constraint keyword, 61, 211
- le0: No carrier - transceiver cable problem message, 338
- Live Upgrade
 - See* Solaris Live Upgrade
- locale file, 55
- locale keyword, 213
- locale.org_dir table, adding entries, 56
- log files
 - begin script output, 159
 - finish script output, 161
 - upgrade installation, 98, 103
- logical AND rule field, 148

M

- Makefile file, 54
- marketing_profile example, 185
- master system
 - See* Web Start Flash
- matching
 - derived profiles, 160
 - order for rules, 149, 177
 - rootdisk values, 215
- memory
 - probe keyword, 220
 - rule keyword, 192, 220
 - swap space size and, 204
- memsize probe keyword, description and values, 220
- memsize rule keyword
 - description and values, 192, 220

- microprocessors
 - probe keywords, 220
 - rule keywords, 189, 220
- mnttab file, 145
- model probe keyword, description and values, 220
- model rule keyword
 - description and values, 192, 220
- modify_install_server, description, 91
- mount command, 91
- mounting
 - begin script caution, 159
 - displaying mounted file systems, 91
 - by Solaris 9 installation, 161
 - remote file systems, 206
- multiple lines in rules files, 147

N

- name server, preconfiguring, 48
- name service, preconfiguring, 48
- names/naming
 - custom_probes file, 172
 - derived profile names, 160
 - host name, 89, 191, 220
 - rules file, 147, 148
 - software groups, 205
 - system model names, 192, 220
 - system platform name determination, 92
- netmask, preconfiguring, 48
- network installation
 - custom JumpStart installation
 - example, 137
 - description, 65
 - preparing, 65
 - requirements, 65
 - using CD media, 81, 85
 - using DHCP, 67
 - using DVD media, 70, 73
- network interface, preconfiguring, 48
- network number, 192, 221
- network probe keyword, description and values, 221
- network rule keyword
 - description and values, 192, 221
- nistbladm command, 56

No carrier - transceiver cable problem message, 338
Not a UFS filesystem message, 338
num_clients profile keyword, 213

O

organization
Solaris 9 DVD, 321
Solaris 9 Installation CD, 322
Solaris 9 Languages CD, 324
Solaris 9 Software CDs, 322
osname probe keyword, 221
osname rule keyword, 193, 221
output files
begin script log, 159
finish script log, 161
upgrade log, 98, 103

P

-p option of check script, 158, 175
package profile keyword, description and values, 213
packages
adding with a finish script, 162
adding with chroot, 163
adding with Live Upgrade, overview, 274
administration file, 159
requirements for an alternate boot environment for Solaris Live Upgrade, 354
requirements for an alternate root (/), 352
requirements when using the JumpStart program and Solaris Live Upgrade, 351
partitioning
examples, 151
excluding disks, 206
profile keyword, 214, 217
partitioning keyword, 214
password, root, 164, 165
Patch Analyzer, 359
patches, 42
adding with a finish script, 162
adding with chroot, 163
adding with Live Upgrade, overview, 274

patches (Continued)
limitations when upgrading to an Update release, 23
paths
check script, 158, 175
permissions
begin scripts, 159
finish scripts, 161
JumpStart directory, 141, 145
pfinstall command, 60, 153
planning
comparison of installation methods, 27
disk space, 33
for Solaris Live Upgrade, 239
initial installation compared to upgrade, 22
installing over the network, 25
overview of tasks, 21
system requirements, 24
system requirements for Live Upgrade, 239
platforms
diskless client, 203
install server setup, 89
matching system attributes and profiles, 149, 177
name determination, 92
probe keywords, 220
rule keywords, 191, 220
system model names, 192, 220
Power Management, 49, 57
preconfiguring system configuration information
advantages, 47
choosing a method, 48
locale using NIS, 54
locale using NIS+, 56
Power Management, 57
using a name service, 49, 54
using sysidcfg file, 49
preparing for installation
custom JumpStart installation, 138, 158
information you need before installing, 38
information you need before upgrading, 42
preconfiguring system information
advantages, 47
methods, 48
preparing the system for install, 37
preparing the system for upgrade, 41

- probe keywords
 - arch, 220
 - disks, 220
 - domainname, 220
 - hostaddress, 220
 - hostname, 220
 - installed, 220
 - karch, 220
 - memsize, 220
 - model, 220
 - network, 221
 - osname, 221
 - rootdisk, 221
 - totaldisk, 221
 - probe rule keyword, description and values, 193
 - processors
 - probe keywords, 220
 - rule keywords, 189, 220
 - profile keywords, 194, 217
 - backup_media, 196, 201
 - boot_device, 202
 - case sensitivity, 194
 - client_arch, 203
 - client_root, 203
 - client_swap, 204
 - cluster
 - description and values, 204, 205
 - examples, 151
 - dontuse
 - description and values, 206
 - usedisk and, 217
 - filesystem
 - description and values, 206
 - examples, 151
 - local file systems, 207
 - remote file systems, 206
 - geo
 - description and values, 209
 - install_type
 - description and values, 210
 - examples, 151
 - requirement, 150, 151
 - isa_bits
 - description and values, 211
 - layout_constraint
 - description and values, 211
 - profile keywords (Continued)
 - locale
 - description and values, 213
 - num_clients, 213
 - package
 - description and values, 213
 - partitioning
 - description and values, 214
 - designating disks, 217
 - examples, 151
 - excluding disks, 206
 - root_device, 215
 - system_type
 - description and values, 216
 - examples, 151
 - usedisk
 - description and values, 217
 - profiles
 - comments in, 151
 - creating, 150
 - derived profiles, 160
 - description, 150
 - examples, 151
 - eng_profile, 185
 - marketing_profile, 185
 - matching systems to, 149, 177
 - naming, 151
 - requirements, 147, 150
 - rule field, 148
 - testing, 60, 155, 156
 - prtvtoc command, creating disk configuration file, 167
- R**
- r option of check script, 158, 176
 - release of Solaris 9 software
 - installed probe keyword, 220
 - installed rule keyword, 191, 220
 - osname probe keyword, 221
 - osname rule keyword, 193, 221
 - remote file systems, mounting, 206
 - requirements
 - custom_probes file, 172
 - disk space, 33
 - memory, 24

- requirements (Continued)
 - network installation
 - servers, 65
 - profiles, 147, 150
 - to use Live Upgrade, 239
 - to use Solaris Web Start, 24
- reset command, 92
- resetting display and terminal after I/O
 - interrupts, 92
- root (/) file systems, package requirements for
 - an alternate root (/), 352
- root (/) file systems, profile example, 138
- root_device keyword, 215
- root environment, customizing with a finish
 - script, 163
- root password, preconfiguring, 48
- root password, setting with a finish script, 164
- rootdisk
 - definition, 215
 - slice value for filesystems, 207
 - value set by JumpStart, 215
- RPC failed: error 5: RPC Timed out message, 144
- RPC Timed out message, 144
- rule_keyword rule field, 148
- rule keywords, 189
 - any
 - description and values, 189, 220
 - arch, 189, 220
 - disksize
 - description and values, 190, 220
 - domainname, 190, 220
 - hostaddress, 190, 220
 - hostname, 189, 191, 220
 - installed
 - description and values, 191, 220
 - karch, 191, 220
 - memsize, 192, 220
 - model, 192, 220
 - network, 192, 221
 - osname, 193, 221
 - probe, 193
 - totaldisk, 194, 221
- rule_value rule field, 148
- rules
 - derived profiles, 160
 - examples, 149
 - field descriptions, 148, 149

- rules (Continued)
 - matching order, 149, 177
 - multiple line rules, 147
 - rootdisk matching rules, 215
 - syntax, 148
 - testing validity, 158, 176
- rules file
 - adding rules, 148
 - comments in, 147
 - creating, 147
 - custom JumpStart example, 186
 - description, 147
 - example, 147
 - multiple line rules, 147
 - naming, 147, 148
 - syntax, 148
 - testing rules, 158
 - validating by using check, 158
 - custom JumpStart example, 187
 - derived profiles and, 160
- rules.ok file
 - creating, 157
 - description, 157
 - matching order for rules, 149, 177

S

- scripts
 - begin scripts, 159, 160, 169
 - Bourne shell scripts in rule fields, 148
 - finish scripts, 161, 165, 169
- security
 - root password, 164, 165
- security policy, preconfiguring, 48
- servers
 - JumpStart directory creation, 141
 - network installation setup with CD media
 - standalone installation, 87
 - network installation setup with DVD media
 - standalone installation, 75
 - requirements for network installation, 65
 - root space, 203
- set_root_pw finish script, 164, 165
- setup_install_server, description, 91
- share command
 - sharing JumpStart directory, 142, 185
- shareable file systems, definition, 226

- shareall command, 142, 185
- sharing JumpStart directory, 142, 185
- SI_CONFIG_DIR variable, 161
- SI_PROFILE environment variable, 160
- site-specific installation programs, 169
- size
 - hard disk
 - probe keywords, 220, 221
 - root space, 203
 - rule keywords, 190, 194, 220, 221
 - space available, 71, 82, 86
 - local file system, 207
 - memory, 192, 220
 - swap space
 - diskless client, 204
 - maximum size, 204
 - profile examples, 138
 - tip window dimensions, 179
- slices
 - customizing file systems for Live Upgrade, 254
 - fileys values, 207
 - guidelines for Live Upgrade, 243
 - probe keyword, 220
 - profile examples, 151
 - rule keyword, 191, 220
- software groups
 - descriptions, 34
 - for profiles, 204, 205
 - profile examples, 151
 - sizes, 35
 - specifying packages, 213
 - upgrading, 23, 60, 205
- Solaris 9 software
 - groups, 204, 205
 - profile examples, 151
 - specifying packages, 213
 - upgrading, 205
 - release or version
 - installed probe keyword, 220
 - installed rule keyword, 191, 220
- Solaris Live Upgrade
 - Activating a boot environment, 282
 - cancel a job, 296
 - changing name of a boot environment, 301
 - commands, 315
 - compare boot environments, 297
 - configuring file systems, 253
 - Solaris Live Upgrade (Continued)
 - copying
 - a shareable file system, 266
 - creating a boot environment, 226, 251
 - deleting a boot environment, 299
 - displaying name of boot environment, 300
 - failed upgrade recovery, 286
 - file system slices, 254
 - installing, 249
 - installing a Web Start Flash archive, 278
 - overview, 226
 - printing to a file, 254
 - starting, 250
 - stopping, 250
 - upgrade task map, 272
 - upgrading a boot environment, 272
 - viewing configuration of boot environments, 305
- Solaris software
 - release or version
 - osname probe keyword, 221
 - osname rule keyword, 193, 221
- Solaris suninstall program
 - compared to other installation methods, 28
 - installing Web Start Flash archives, 119
 - ways to run, 101
- Solaris Web Start
 - command-line interface (CUI), 93
 - compared to other installation methods, 27
 - graphical user interface (GUI), 93
 - installation program, 93
 - installing Web Start Flash archives, 117
 - ways to run, 93
- standalone systems
 - custom JumpStart installation example, 134
 - profile examples, 151
- starting
 - check script, 157, 158
 - tftpd daemon, 342
- status, display boot environment, 293
- stty command, 37, 41, 179
- subnet
 - boot server creation on with CD media, 85
 - boot server creation on with DVD media, 73
- SUNWCall group, 204
- SUNWCprog group, 204
- SUNWCreq group, 204

- SUNWCuser group, 204
- SUNWCXall group, 204
- swap file systems
 - customizing for Live Upgrade (character interface), 256
 - customizing for Live Upgrade (command line), 262
 - diskless client swap space, 204
 - memory size and, 204
 - overview for Solaris Live Upgrade, 226
 - profile examples, 138
 - size determination, 204
- sysidcfg file
 - guidelines and requirements, 49
 - how to create, 53
 - keywords, 50
 - syntax, 50
- system information, displaying, 92
- system_type profile keyword
 - description and values, 216
 - examples, 151

T

- terminal type, preconfiguring, 48
- testing
 - profiles, 60, 153, 155, 156
 - validating custom_probes files
 - testing custom_probes, 176
 - using check, 175
 - validating rules file
 - using check, 176
 - validating rules files
 - custom JumpStart example, 187
 - derived profiles and, 160
 - testing rules, 158
 - using check, 157, 158
- ftpsd daemon, 341
- time and date, preconfiguring, 48
- time zone, preconfiguring, 48
- tip line connection and custom JumpStart installation, 179
- totaldisk probe keyword, 221
- totaldisk rule keyword, 194, 221
- transceiver cable problem message, 338
- troubleshooting
 - booting from wrong server, 341

- troubleshooting (Continued)
 - general installation problems
 - booting the system, 341

U

- UFS, 145
- uname command, 92
- Unknown client error message, 337
- upgrade
 - compared to initial installation, 23, 60
 - comparison of methods, 27
 - disk space recommendations, 33
 - methods, 23, 60
 - overview of tasks, 21
 - system requirements, 24
 - to a Solaris Update release, 359
 - with custom JumpStart, 60
 - with Disk Space Reallocation, 61
 - with Solaris Live Upgrade, 272
- upgrade installation
 - custom JumpStart installation, 177
 - failed upgrade, 343
 - failed upgrade recovery for Solaris Live Upgrade, 286
 - log file, 98, 103
 - profile keywords, 205, 210, 214
- upgrade_log file, 98, 103
- upgrading, with Live Upgrade, overview, 231
- upgrading cleanup, 98, 104
- usedisk profile keyword, description and values, 217

V

- validating
 - custom_probes file
 - testing, 176
 - using check, 176
 - profiles, 155
 - rules files
 - custom JumpStart example, 187
 - derived profiles and, 160
 - testing rules, 158
 - using check, 157, 158, 176
 - /var/sadm/begin.log file, 159

- /var/sadm/finish.log file, 161
- /var/sadm/install_data/upgrade_log file, 103
- /var/yp/make command, 56
- /var/yp/Makefile, 54
- variables
 - SI_CONFIG_DIR, 161
 - SI_PROFILE, 160
 - SYS_MEMSIZE, 155
- version of Solaris 9 software, installed rule keyword, 191
- version of Solaris software
 - installed probe keyword, 220
 - installed rule keyword, 220
 - osname probe keyword, 221
 - osname rule keyword, 193, 221
- volcheck command, 145
- Volume Manager, copying, 145

- Web Start Flash (Continued)
 - master system
 - description, 108
 - how to install, 114
 - peripheral devices, 109
 - wrapping lines in rules files, 147

W

- WARNING: CHANGE DEFAULT BOOT DEVICE, 342
- WARNING: clock gained xxx days message, 338
- Web Start Flash
 - archive sections description, 123
 - compared to other installation methods, 29
 - description, 107
 - flar command
 - description, 128
 - to combine archives, 130
 - to extract archive information, 129
 - to split archives, 129
 - flarcreate command, 126
 - how to create an archive, 114
 - installing
 - how to install, 117
 - installation methods, 111
 - installing on Live Upgrade boot environment, 278
 - keywords
 - custom JumpStart, 196
 - section_begin and section_end, 124
 - section identification, 124
 - layered archives, 112