Contents

Preface  17

1 Planning for Solaris Installation or Upgrade (Topics)  21

2 Planning for a Solaris Installation or Upgrade (Overview)  23
   Task Map: Installing or Upgrading the Solaris Software  23
   Initial Installation or Upgrade?  24
      Initial Installation  25
      Upgrade  25
   System Requirements  27
      Memory Requirement  27
      Requirements When Using the Solaris 9 Installation CD  27
   Installing From the Network or From DVD or CDs?  29
   Using DVD Media  30
   Sun ONE Application Server Post-Installation Configuration  30
   x86: Change in Default Boot-Disk Partition Layout  30
   x86: Accessing the Solaris 9 Device Configuration Assistant and PXE  31

3 Choosing a Solaris Installation Method (Overview)  33
   Solaris Web Start Installation Program  33
   Solaris suninstall Program  34
   Custom JumpStart Installation Method  34
   Solaris Flash Installation Feature  35
   Solaris Live Upgrade Method  36
   SPARC: Factory JumpStart Installation Method  36
4 Preparing for Solaris Installation or Upgrade (Topics) 37

5 Guidelines for Allocating Disk Space and Swap Space (Planning) 39
   Allocating Disk Space 39
       General Disk Space Planning and Recommendations 39
       Disk Space Recommendations for Software Groups 40
   SPARC: How Swap Space Is Allocated 41

6 Gathering Information Before Installation or Upgrade (Planning) 43
   Checklist for Installation 43
   Worksheet for Installation 44
   Checklist for Upgrading 49
   Worksheet for Upgrading 50

7 Preconfiguring System Configuration Information (Tasks) 55
   Advantages of Preconfiguring System Configuration Information 55
   Ways to Preconfigure System Configuration Information 56
   Preconfiguring With the sysidcfg File 57
       Syntax Rules for the sysidcfg File 58
       sysidcfg File Keywords 59
           ▼ To Create a sysidcfg Configuration File 61
   Preconfiguring With the Name Service 63
       ▼ To Preconfigure the Locale Using NIS 63
       ▼ To Preconfigure the Locale Using NIS+ 65
   SPARC: Preconfiguring Power Management Information 66

8 Upgrading the Solaris Operating Environment (Planning) 69
   Upgrading 69
       Solaris Upgrade Methods 70
   Using Solaris Live Upgrade 71
   Using Custom JumpStart to Upgrade 71
   Upgrading With Disk Space Reallocation 72
   Backing Up Systems Before Upgrading 73
13 Preparing to Install From the Network (Command Reference)  123

14 Using the Solaris Web Start Program (Tasks)  125
   Solaris Web Start Program GUI or CLI  125
   SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program 126
      ▼ SPARC: Task Map: Performing a Solaris Web Start Installation  127
      ▼ SPARC: To Perform an Installation or Upgrade With the Solaris Web Start Program 127
   x86: Performing an Installation or Upgrade With the Solaris Web Start Program  131
      ▼ x86: Task Map: Performing a Solaris Web Start Installation  131
      ▼ x86: To Perform an Installation or Upgrade With the Solaris Web Start Program 132
   Solaris Web Start Post-Installation and Upgrade Tasks  138
      ▼ To Correct Local Modifications After Upgrading  139

15 Using the Solaris suninstall Program (Tasks)  141
   Solaris suninstall Program  141
   SPARC: Performing an Installation or Upgrade With the Solaris suninstall Program  142
      ▼ SPARC: Task Map: Performing a Solaris suninstall Program Installation  142
      ▼ SPARC: To Perform an Installation or Upgrade With the Solaris suninstall Program 143
   x86: Performing an Installation or Upgrade With the Solaris suninstall Program  145
      ▼ x86: suninstall Installation Task Map  145
      ▼ x86: To Perform an Installation or Upgrade With the Solaris suninstall Program  146
   Solaris suninstall Program Post-Upgrade Task  151
      ▼ To Correct Local Modifications After Upgrading  151

16 Solaris Flash Installation Feature (Topics)  153

17 Solaris Flash (Overview and Planning)  155
   Solaris Flash Introduction  155
      Installing Clone Systems With an Initial Installation  155
      Updating Clone Systems With a Solaris Flash Differential Archive  157
   Planning Your Solaris Flash Installation  160
18 Creating Solaris Flash Archives (Tasks) 169

Task Map: Creating Solaris Flash Archives 169

Installing the Master System 170

▼ To Install the Master System for an Initial Installation 171

Creating Customization Scripts 171

▼ To Create a Precreation Script 172

▼ To Create a Predeployment Script 173

▼ To Create a Postdeployment Script 174

▼ To Create a Reboot Script 175

Creating a Solaris Flash Archive 175

▼ To Create a Solaris Flash Archive for an Initial Installation 176

▼ To Update a Master Image and Create a Differential Archive 180

19 Installing and Administering Solaris Flash Archives (Tasks) 185

References to Procedures for Installing Solaris Flash Archives 185

Administering Solaris Flash Archives 186

Splitting a Solaris Flash Archive 186

Merging a Solaris Flash Archive 187

Extracting Information From an Archive 188

20 Solaris Flash (Reference) 191

Solaris Flash Archive Section Descriptions 191

Solaris Flash Keywords 193

General Keywords 193

Identification Section Keywords 194

User-Defined Section Keywords 197

Solaris Flash flar create Command 198

flar create 198
21 Custom JumpStart Installation (Topics) 203

22 Custom JumpStart (Overview) 205
   Custom JumpStart Introduction 205
   How the JumpStart Program Installs Solaris Software 206

23 Preparing Custom JumpStart Installations (Tasks) 211
   Task Map: Preparing Custom JumpStart Installations 211
   Creating a Profile Server for Networked Systems 213
      ▼ To Create a JumpStart Directory on a Server 213
      ▼ To Allow All Systems Access to the Profile Server 215
   Creating a Profile Diskette for Standalone Systems 217
      ▼ SPARC: To Create a Profile Diskette 217
      ▼ x86: To Create a Profile Diskette 219
   Creating the rules File 221
      Syntax of the rules File 222
      ▼ To Create a rules File 222
      rules File Example 224
   Creating a Profile 225
      Syntax of Profiles 225
      ▼ To Create a Profile 225
      Profile Examples 226
   Testing a Profile 229
      ▼ To Create a Temporary Solaris 9 Environment to Test a Profile 230
      ▼ To Test a Profile 231
      Profile Test Examples 232
   Validating the rules File 233
      ▼ To Validate the rules File 233

24 Using Optional Custom JumpStart Features (Tasks) 235
   Creating Begin Scripts 235
      Important Information About Begin Scripts 236
      Creating Derived Profiles With a Begin Script 236
   Creating Finish Scripts 237
      Important Information About Finish Scripts 237
      ▼ To Add Files With a Finish Script 238
      Adding Packages or Patches With a Finish Script 238
Customizing the Root Environment With a Finish Script 239
Setting a System’s Root Password With a Finish Script 240
Installing Software With Web Start Installation Programs With Finish Scripts 241
Creating a Compressed Configuration File 242
  ▼ To Create a Compressed Configuration File 242
  Compressed Configuration File Example 243
Creating Disk Configuration Files 243
  ▼ SPARC: To Create a Disk Configuration File 243
  SPARC: Disk Configuration File Example 244
  ▼ x86: To Create a Disk Configuration File 245
  x86: Disk Configuration File Example 246
Using a Site-Specific Installation Program 248

25 Creating Custom Rule and Probe Keywords (Tasks) 249
  Probe Keywords 249
  Creating a custom_probes File 250
    Syntax of the custom_probes File 250
    Syntax of Function Names in custom_probes 251
    ▼ To Create a custom_probes File 251
    Examples of a custom_probes File and Keyword 251
  Validating the custom_probes File 253
    ▼ To Validate the custom_probes File 253

26 Performing a Custom JumpStart Installation (Tasks) 255
  SPARC: Performing a Custom JumpStart Installation 255
    SPARC: Task Map: Setting Up a System for a Custom JumpStart Installation 255
    ▼ To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation 257
    ▼ SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program 258
  x86: Performing a Custom JumpStart Installation 260
    x86: Task Map: Setting Up a System for a Custom JumpStart Installation 260
    ▼ x86: To Perform an Installation or Upgrade With the Custom JumpStart Program 262
27 Installing With Custom JumpStart (Examples) 267
   Sample Site Setup 267
   Create an Install Server 269
   x86: Create a Boot Server for Marketing Systems 270
   Create a JumpStart Directory 271
   Share the JumpStart Directory 271
   SPARC: Create the Engineering Group’s Profile 271
   x86: Create the Marketing Group’s Profile 272
   Update the rules File 272
   Validate the rules File 273
   SPARC: Set Up Engineering Systems to Install From the Network 273
   x86: Set Up Marketing Systems to Install From the Network 274
   SPARC: Boot the Engineering Systems and Install Solaris 9 Software 275
   x86: Boot the Marketing Systems and Install Solaris 9 Software 275

28 Custom JumpStart (Reference) 277
   Rule Keywords and Values 277
   Profile Keywords and Values 282
      Profile Keywords at a Glance 282
      Profile Keyword Descriptions and Examples 283
   Custom JumpStart Environment Variables 308
   Probe Keywords and Values 311

29 Solaris Live Upgrade (Topics) 313

30 Solaris Live Upgrade (Overview) 315
   Solaris Live Upgrade Introduction 315
      Solaris Live Upgrade Process 316

31 Solaris Live Upgrade (Planning) 329
   Solaris Live Upgrade System Requirements 329
   Solaris Live Upgrade Disk Space Requirements 330
   Managing Packages and Patches With Solaris Live Upgrade 330
      Required Packages 331
      Checking System Patch Levels 332
   Guidelines for Creating File Systems With the lucreate Command 332
Managing Packages and Patches With Solaris Live Upgrade 364

▼ To Upgrade an Operating System Image on a Boot Environment (Character Interface) 365

▼ To Upgrade an Operating System Image on a Boot Environment (Command-Line Interface) 366

▼ To Upgrade an Operating System Image From Multiple CDs (Command-Line Interface) 366

▼ To Create a Profile to be Used by Solaris Live Upgrade 368

▼ To Test a Profile to Be Used By Solaris Live Upgrade 370

▼ To Upgrade With a Profile by Using Solaris Live Upgrade (Command-Line Interface) 371

Installing Solaris Flash Archives on a Boot Environment 372

Upgrading Metadevices and Volumes 373

▼ To Install a Solaris Flash Archive on a Boot Environment (Character Interface) 374

▼ To Install a Solaris Flash Archive on a Boot Environment (Command-Line Interface) 375

▼ To Install a Solaris Flash Archive With a Profile (Command-Line Interface) 376

▼ To Install a Solaris Flash Archive With a Profile Keyword (Command-Line Interface) 377

Activating a Boot Environment 378

▼ x86: (Optional) Update Boot Diskette Before Activating 379

▼ To Activate a Boot Environment (Character Interface) 379

▼ To Activate a Boot Environment (Command-Line Interface) 380

▼ To Activate a Boot Environment and Synchronize Files (Command-Line Interface) 381

Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface) 382

▼ To Fall Back Despite Successful New Boot Environment Creation 383

▼ SPARC: To Fall Back From a Failed Boot Environment Activation 383

▼ SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image 384

▼ x86: To Fall Back With Boot Environments on Different Disks 385

▼ x86: To Fall Back With Boot Environments on the Same Disk 386

34 Maintaining Solaris Live Upgrade Boot Environments (Tasks) 389

Overview of Solaris Live Upgrade Maintenance 390

Displaying the Status of All Boot Environments 391

▼ To Display the Status of All Boot Environments (Character Interface) 391
To Display the Status of All Boot Environments (Command-Line Interface) 392

Updating a Previously Configured Boot Environment 392
  ▼ To Update a Previously Configured Boot Environment (Character Interface) 393
  ▼ To Update a Previously Configured Boot Environment (Command-Line Interface) 393

Canceling a Scheduled Create, Upgrade, or Copy Job 394
  ▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Character Interface) 394
  ▼ To Cancel a Scheduled Create, Upgrade, or Copy Job (Command-Line Interface) 395

Comparing Boot Environments 395
  ▼ To Compare Boot Environments (Character Interface) 395
  ▼ To Compare Boot Environments (Command-Line Interface) 396

Deleting an Inactive Boot Environment 397
  ▼ To Delete an Inactive Boot Environment (Character Interface) 397
  ▼ To Delete an Inactive Boot Environment (Command-Line Interface) 397

Displaying the Name of the Active Boot Environment 398
  ▼ To Display the Name of the Active Boot Environment (Character Interface) 398
  ▼ To Display the Name of the Active Boot Environment (Command-Line Interface) 398

Changing the Name of a Boot Environment 399
  ▼ To Change the Name of an Inactive Boot Environment (Character Interface) 399
  ▼ To Change the Name of an Inactive Boot Environment (Command-Line Interface) 399

Adding or Changing a Description Associated With a Boot Environment Name 400
  ▼ To Add or Change a Description for a Boot Environment Name With Text 400
  ▼ To Add or Change a Description for a Boot Environment Name With a File 401
  ▼ To Determine a Boot Environment Name From a Text Description 401
  ▼ To Determine a Boot Environment Name From a Description in a File 402

Viewing the Configuration of a Boot Environment 403
  ▼ To View the Configuration of Each Inactive Boot Environment (Character Interface) 403
  ▼ To View the Configuration of a Boot Environment (Command-Line Interface) 403
35 Solaris Live Upgrade (Examples) 405
   Example of Upgrading With Solaris Live Upgrade (Command-Line Interface) 405
      Install Live Upgrade on the Active Boot Environment 405
      Create a Boot Environment 406
      Upgrade the Inactive Boot Environment 406
      Check If Boot Environment Is Bootable 406
      Activate the Inactive Boot Environment 407
      Fall Back to the Source Boot Environment 407
   Example of Upgrading Using Solaris Live Upgrade (Character Interface) 408
      Install Live Upgrade on the Active Boot Environment 408
      Create a Boot Environment 409
      Upgrade the Inactive Boot Environment 411
      Activate the Inactive Boot Environment 411

36 Solaris Live Upgrade (Command Reference) 413

37 Solaris Software Reference (Topics) 415

38 Organization of Solaris 9 Media (Reference) 417
   SPARC: Solaris 9 Media 417
      SPARC: Directory Organization of Solaris 9 Media 418
         SPARC: Solaris 9 Installation CD Directory Structure 420
   x86: Solaris 9 Media 424
      x86: Directory Organization of Solaris 9 Media 425
         x86: Solaris 9 Installation CD Directory Structure 427
         x86: Solaris 9 Software x86 Platform Edition CDs Directory Structures 427
         x86: Solaris 9 x86 Platform Edition Languages CD Directory Structures 429
Preface

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

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

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

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

Who Should Use This Book

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

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

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

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Administration Guide: Basic Administration</td>
<td>Describes how to back up system files</td>
</tr>
<tr>
<td>Solaris 9 4/03 Release Notes</td>
<td>Describes any bugs, known problems, software that is being discontinued, and patches that are related to the Solaris release</td>
</tr>
<tr>
<td>SPARC: Solaris 9 Sun Hardware Platform Guide</td>
<td>Contains information about supported hardware</td>
</tr>
<tr>
<td>Solaris 9 4/03 Package List</td>
<td>Lists and describes the packages in the Solaris 9 4/03 operating environment</td>
</tr>
</tbody>
</table>

Accessing Sun Documentation Online

The docs.sun.com 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>
<thead>
<tr>
<th>Typeface or Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>The names of commands, files, and directories; on-screen computer output</td>
<td>Edit your .login file. Use 1s -a to list all files. machine_name% you have mail.</td>
</tr>
</tbody>
</table>
TABLE P–2 Typographic Conventions  (Continued)

<table>
<thead>
<tr>
<th>Typeface or Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>What you type, contrasted with on-screen computer output</td>
<td>machine_name% su Password:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Command-line placeholder: replace with a real name or value</td>
<td>To delete a file, type rm filename.</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Book titles, new words, or terms, or words to be emphasized.</td>
<td>Read Chapter 6 in User’s Guide. These are called class options. You must be root to do this.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell prompt</td>
<td>machine_name%</td>
</tr>
<tr>
<td>C shell superuser prompt</td>
<td>machine_name#</td>
</tr>
<tr>
<td>Bourne shell and Korn shell prompt</td>
<td>$</td>
</tr>
<tr>
<td>Bourne shell and Korn shell superuser prompt</td>
<td>#</td>
</tr>
</tbody>
</table>
Planning for Solaris Installation or Upgrade (Topics)

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

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 2</td>
<td>Provides information about decisions that you need to make before you install or upgrade.</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Provides details about the different Solaris installation technologies to help you choose which method is best for your environment.</td>
</tr>
</tbody>
</table>
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 23
- “Initial Installation or Upgrade?” on page 24
- “System Requirements” on page 27
- “Installing From the Network or From DVD or CDs?” on page 29
- “Using DVD Media” on page 30
- “Sun ONE Application Server Post-Installation Configuration” on page 30
- “x86: Change in Default Boot-Disk Partition Layout” on page 30
- “x86: Accessing the Solaris 9 Device Configuration Assistant and PXE” on page 31

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

Task Map: Installing or Upgrading the Solaris Software

The following task map is an overview of the steps necessary to install or upgrade the Solaris operating environment. Use this task map to identify all of the decisions that you need to make to complete the most efficient installation for your environment.
TABLE 2-1 Task Map: Installing or Upgrading the Solaris Software

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose initial installation or upgrade.</td>
<td>Decide if you want to perform an initial installation or an upgrade.</td>
<td>“Initial Installation or Upgrade?” on page 24</td>
</tr>
<tr>
<td>Review system requirements.</td>
<td>Determine if your system meets the minimum requirements to install or upgrade.</td>
<td>“System Requirements” on page 27</td>
</tr>
<tr>
<td>Choose an installation method.</td>
<td>The Solaris operating environment provides several methods for installation or upgrade. Choose the installation method that is most appropriate for your environment.</td>
<td>Chapter 3</td>
</tr>
<tr>
<td>Plan and allocate disk space and swap space.</td>
<td>Allocate disk space on your system for the components of the Solaris operating environment that you want to install. Determine the appropriate swap space layout for your system.</td>
<td>Chapter 5</td>
</tr>
<tr>
<td>Choose an installation location.</td>
<td>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.</td>
<td>“Installing From the Network or From DVD or CDs?” on page 29</td>
</tr>
<tr>
<td>Gather information about your system.</td>
<td>Use the checklist and complete the worksheet to collect all of the information that you need to install or upgrade.</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>(Optional) Preconfigure system information.</td>
<td>You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>(Optional) Prepare to install the Solaris software from the network.</td>
<td>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.</td>
<td>Chapter 12</td>
</tr>
<tr>
<td>(Upgrade only) Perform the pre-upgrade tasks.</td>
<td>Back up your system, determine if you can upgrade with disk space reallocation, and search for patches that a Solaris Update release might override.</td>
<td>Chapter 8</td>
</tr>
<tr>
<td>Install or upgrade.</td>
<td>Use the Solaris installation method that you chose to install or upgrade the Solaris software.</td>
<td>The chapter or chapters that provide detailed instructions for the installation method</td>
</tr>
</tbody>
</table>

**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:

```bash
$ uname -a
```

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

---

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

---
## Solaris Upgrade Methods

### TABLE 2-2 SPARC: Solaris Upgrade Methods

<table>
<thead>
<tr>
<th>Current Solaris Operating Environment</th>
<th>Solaris Upgrade Methods</th>
</tr>
</thead>
</table>
| Solaris 2.6, Solaris 7, Solaris 8, Solaris 9 | - Solaris™ Web Start program  
- Solaris suninstall program  
- Custom JumpStart™ method  
- Solaris Live Upgrade |

### TABLE 2-3 x86: Solaris Upgrade Methods

<table>
<thead>
<tr>
<th>Current Solaris Operating Environment</th>
<th>Solaris Upgrade Methods</th>
</tr>
</thead>
</table>
| Solaris 2.6                           | Installing from DVD media or a net installation image:  
- Solaris Web Start program  
- Solaris suninstall program  
- Custom JumpStart method  
Installing from CD media:  
- Solaris suninstall program  
- Custom JumpStart method |
| Solaris 7                             | Installing from DVD media or a net installation image:  
- Solaris Web Start program  
- Solaris suninstall program  
- Custom JumpStart method  
- Solaris Live Upgrade  
Installing from CD media:  
- Solaris suninstall program  
- Custom JumpStart method  
- Solaris Live Upgrade |
| Solaris 8, Solaris 9                  | Installing from DVD or CD media or a net installation image:  
- Solaris Web Start program  
- Solaris suninstall program  
- Custom JumpStart method  
- Solaris Live Upgrade |

**Note** – For limitations on upgrading using Solaris Live Upgrade, see “Solaris Live Upgrade System Requirements” on page 329
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 471.

### 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 are installing or upgrading by using the Solaris 9 Installation CD, there are special requirements for SPARC slices and x86 fdisk partitions. When you are installing from a DVD or a net installation image, these requirements are not necessary.
### TABLE 2-4 Solaris 9 Installation CD Requirements

<table>
<thead>
<tr>
<th>Platform</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice requirements for upgrading</td>
<td>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.</td>
</tr>
</tbody>
</table>
| x86 systems fdisk partition requirements | When you use the Solaris 9 Installation CD, the Solaris Web Start program requires two fdisk partitions on the system disk to perform an installation or upgrade.  
  - Solaris fdisk partition  
    This is the typical Solaris fdisk partition. If you do not have a Solaris fdisk partition on your system, the Solaris Web Start program prompts you to create one.  
    Caution: If you change the size of an existing fdisk partition, all data on that partition is automatically deleted. Back up your data before you create a Solaris fdisk partition.  
  - x86 boot fdisk partition  
    This is a 10-Mbyte fdisk partition that enables the x86 architecture to boot the minroot that is placed on the newly created swap slice that is located on the Solaris fdisk partition.  
    Caution: Do not create the x86 boot partition manually.  
    The Solaris Web Start installation program creates the x86 boot partition by removing 10 Mbytes from the Solaris fdisk partition. By allowing the installation program to create the x86 boot partition, you prevent any existing fdisk partitions from being altered.  
  
  **Note** – If you install or upgrade a system that has a Service partition, the Solaris Web Start installation program preserves the Service partition and creates the Solaris and x86 boot fdisk partitions. For more information on preserving a Service partition, see “x86: Change in Default Boot-Disk Partition Layout” on page 30. |
| x86 system upgrade limitations   | When you use the Solaris 9 Installation CD, you cannot use the Solaris Web Start program to upgrade from the Solaris 2.6 or Solaris 7 operating environments. The Solaris 9 Installation CD requires a separate 10-Mbyte x86 boot partition that was not required in the Solaris 2.6 or Solaris 7 releases. You must use the Solaris Web Start program from a DVD or a net installation image, or use the Solaris suninstall program or custom JumpStart to upgrade. |
TABLE 2–4 Solaris 9 Installation CD Requirements  (Continued)

<table>
<thead>
<tr>
<th>Platform</th>
<th>Requirements</th>
</tr>
</thead>
</table>
| x86 systems logical block addressing requirement | Do not use the Solaris 9 Installation CD unless your system can boot across the 1024–cylinder limit. Logical block addressing (LBA) enables the machine to boot beyond the 1024–cylinder limit and across Solaris disk slices. Use the Solaris 9 Installation CD when your system’s BIOS and SCSI driver for the default boot disk supports LBA.  
To determine if your system supports LBA, type:  

```
# prtconf -pv | grep -i lba
```

If the BIOS and SCSI driver for the default boot disk support LBA, the following message appears.  

```
lba-access-ok:
```

If the SCSI driver for the default boot disk does not support LBA, the following message appears.  

```
no-bef-lba-access
```

If the BIOS and SCSI driver for the default boot disk do not support LBA, use the Solaris 9 DVD or a net installation image to install or upgrade. |

---

### Installing From the Network or From DVD or CDs?

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

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

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

Installing the Solaris software from the network requires initial setup. For 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

Sun ONE Application Server Post-Installation Configuration

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

x86: Change in Default Boot-Disk Partition Layout

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

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

The new default includes the following partitions.

- First partition – Service partition (existing size on system)
- Second partition – x86 boot partition (approximately 11 Mbytes)
Third partition – Solaris operating environment partition (remaining space on the boot disk)

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

x86: Accessing the Solaris 9 Device Configuration Assistant and PXE

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

- By booting from the Solaris 9 DVD, Solaris 9 Installation CD, or Solaris 9 Software 1 of 2 CD. Your system’s BIOS must support booting from a DVD or CD.
- By booting from a net installation image of Solaris Software with Preboot Execution Environment (PXE). PXE enables you to boot a system directly from the network without using the boot diskette. The system must support PXE. Enable the system to use PXE by using the system’s BIOS setup tool or the network adapter’s configuration setup Tool. For more information on configuring your system for using PXE network boot, see “x86: PXE Network Boot” on page 477.
- By booting from a diskette. You can copy the Device Configuration Assistant software to a diskette by the following methods:
  - Copy the software to a diskette from the Solaris 9 x86 Platform Edition DVD or Solaris 9 Software 2 of 2 x86 Platform Edition CD by using the copy procedure in Appendix E.
  - Download and copy the software to a diskette from the Solaris Developer Connection at http://soldc.sun.com/support/drivers/dca_diskettes
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 33
- “Solaris suninstall Program” on page 34
- “Custom JumpStart Installation Method” on page 34
- “Solaris Flash Installation Feature” on page 35
- “Solaris Live Upgrade Method” on page 36
- “SPARC: Factory JumpStart Installation Method” on page 36

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 Solaris 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 Solaris 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.

Solaris Flash Installation Feature

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

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

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

If you want to update a previously installed clone system, you can install changes with a differential archive. With the custom JumpStart installation method, the differential archive can install archives that contain only the differences between two system images.

If you have many different configurations that you want to install on your systems, you need a Solaris Flash archive for each system. Solaris Flash archives are large files and require a significant amount of disk space. Also, after you create a Solaris 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. Alternatively, you can accomplish system-specific customizations by using a JumpStart finish script or an embedded Flash postdeployment script.
For an overview of Solaris Flash installation feature including installing an archive with an initial installation or an update, and using scripts, see Chapter 17.

For detailed instructions on installing an archive with custom JumpStart, see “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 257.

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 Solaris 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.

SPARC: 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. You cannot use the JumpStart installation method on x86 based systems.
Preparing for Solaris Installation or Upgrade (Topics)

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

<table>
<thead>
<tr>
<th>Chapter 5</th>
<th>Provides guidelines to help you plan the disk space that you need to install or upgrade the Solaris operating environment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 6</td>
<td>Provides a comprehensive list of the information about your system that you need to obtain before you begin the Solaris installation.</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Provides instructions for using the <code>sysidcfg</code> file to specify system information during the Solaris installation.</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Provides detailed requirements for upgrading the Solaris operating environment.</td>
</tr>
</tbody>
</table>
Guidelines for Allocating Disk Space and Swap Space (Planning)

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

- “Allocating Disk Space” on page 39
- “SPARC: How Swap Space Is Allocated” on page 41

Allocating Disk Space

Before you install the Solaris 9 software, you can determine if your system has enough disk space by doing some high-level planning. If you plan your disk space needs before you install, you can add more disks to your system, if you need them, before you install the Solaris 9 software. If you are upgrading, review “Upgrading With Disk Space Reallocation” on page 72.

General Disk Space Planning and Recommendations

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

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

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

If you are using Solaris Live Upgrade, see “Solaris Live Upgrade Disk Space Requirements” on page 330.

### Disk Space Recommendations for Software Groups

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

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

When you are installing the Solaris software, you can choose to add or remove packages from the Solaris software group that you selected. When you are selecting which packages to add or remove, you need to know about software dependencies and how the Solaris software is packaged.
The following table lists the Solaris software groups and the recommended amount of disk space that you need to install each group.

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

<table>
<thead>
<tr>
<th>Software Group</th>
<th>Recommended Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire Solaris Software Group Plus OEM Support</td>
<td>2.9 Gbytes</td>
</tr>
<tr>
<td>Entire Solaris Software Group</td>
<td>2.7 Gbytes</td>
</tr>
<tr>
<td>Developer Solaris Software Group</td>
<td>2.4 Gbytes</td>
</tr>
<tr>
<td>End User Solaris Software Group</td>
<td>2.0 Gbytes</td>
</tr>
<tr>
<td>Core System Support Software Group</td>
<td>2.0 Gbytes</td>
</tr>
</tbody>
</table>

SPARC: How Swap Space Is Allocated

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

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

- For the Solaris Web Start and Solaris suninstall programs, you can customize the disk layout in cylinder mode and manually assign the swap slice to the desired location.
- For the Solaris custom JumpStart installation program, you can assign the swap slice in the profile file. For more information on the Solaris Custom JumpStart profile file, see “Creating a Profile” on page 225.

For an overview of the swap space, see “Configuring Additional Swap Space (Tasks)” in System Administration Guide: Basic Administration.
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 43
- “Worksheet for Installation” on page 44
- “Checklist for Upgrading” on page 49
- “Worksheet for Upgrading” on page 50

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 27.

- Verify that you have enough disk space. For more information, refer to Chapter 5.

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

- Review the Solaris 9 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>
<thead>
<tr>
<th>TABLE 6–1 Installation Worksheet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Info Needed to Install</strong></td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>DHCP</td>
</tr>
<tr>
<td>Host Name</td>
</tr>
<tr>
<td>IP Address</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Subnet</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>IPv6</td>
</tr>
<tr>
<td>Info Needed to Install</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Kerberos</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Name Service</td>
</tr>
<tr>
<td>Domain Name</td>
</tr>
<tr>
<td>NIS+ and NIS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DNS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Info Needed to Install</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
</tbody>
</table>
| LDAP                   | Provide the following information about your LDAP profile.  
                          | Profile Name:  
                          | Profile Server:  
                          | If you specify a proxy credential level in your LDAP profile, gather this information.  
                          | Proxy-bind distinguished name: | Proxy-bind password: |
| Default Route          | Do you want to specify a default route IP address or let the Solaris Web Start installation program or Solaris suninstall program find one?  
                          | The default route provides a bridge that forwards traffic between two physical networks. An IP address is a unique number that identifies each host on a network.  
                          | You have the following choices:  
                          | You can specify the IP address. An /etc/defaultrouter file is created with the specified IP address. When the system is rebooted, the specified IP address becomes the default route.  
                          | You can let the software detect an IP address.  
                          | The Solaris suninstall program detects an IP address when the system is rebooted.  
                          | The Solaris Web Start installation program can detect a default route. However, the system must be on a subnet that has a router that advertises itself by using the ICMP router discovery protocol.  
                          | You can choose None if you do not have a router or do not want the software to detect an IP address at this time. The software automatically tries to detect an IP address on reboot.  
                          | Host IP address: |
| Time Zone              | How do you want to specify your default time zone? | Geographic region  
                          | Offset from GMT  
                          | Time zone file |
# TABLE 6–1 Installation Worksheet  (Continued)

<table>
<thead>
<tr>
<th>Info Needed to Install</th>
<th>Description/Example</th>
<th>Enter Your Answers Here:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locales</td>
<td>For which geographic regions do you want to install support?</td>
<td></td>
</tr>
<tr>
<td>SPARC: Power Management (only available on SPARC systems that support Power Management)</td>
<td>Do you want to use Power Management? <strong>Note</strong> – If your system has Energy Star version 3 or later, you are not prompted for this information.</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Proxy Server Configuration (only available in the Solaris Web Start program)</td>
<td>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.</td>
<td>Direct Connection/Proxy Server</td>
</tr>
<tr>
<td></td>
<td>Host:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port:</td>
<td></td>
</tr>
<tr>
<td>Automatic reboot or CD/DVD ejection</td>
<td>Reboot automatically after software installation? Eject CD/DVD automatically after software installation?</td>
<td>Yes/No Yes/No</td>
</tr>
<tr>
<td>Software Group</td>
<td>Which Solaris Software Group do you want to install?</td>
<td>Entire Plus OEM Entire Developer End User Core</td>
</tr>
<tr>
<td>Custom Package Selection</td>
<td>Do you want to add or remove software packages from the Solaris Software Group that you install? <strong>Note</strong> – When you select which packages to add or remove, you need to know about software dependencies and how Solaris software is packaged.</td>
<td></td>
</tr>
<tr>
<td>SPARC: 64-bit</td>
<td>Do you want to install support for 64–bit applications?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Select Disks</td>
<td>On which disks do you want to install the Solaris software? Example: c0t0d0</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 6–1 Installation Worksheet  (Continued)

<table>
<thead>
<tr>
<th>Info Needed to Install</th>
<th>Description/Example</th>
<th>Enter Your Answers Here:</th>
</tr>
</thead>
</table>
| x86: fdisk partitioning| Do you want to create, delete, or modify a Solaris fdisk partition?  
Each disk selected for file system layout must have a Solaris fdisk partition. Only one x86 Boot partition is allowed per system.  
If your system currently has a Service partition, the Solaris Web Start installation program preserves the Service partition by default. If you do not want to preserve the Service partition, you must customize the fdisk partitions. For more information on preserving a Service partition, see “x86: Change in Default Boot-Disk Partition Layout” on page 30  
Select Disks for fdisk Partition Customization?  
Customize fdisk partitions?                                                                 | Yes/No                    |
| Preserve Data          | Do you want to preserve any data that exists on the disks where you are installing the Solaris software?                                                                                                               | Yes/No                    |
| Auto-layout            | Do you want the installation program to automatically lay out file systems on your disks?                                                                                                                             | Yes/No                    |
| File Systems           | If yes, which file systems should be used for auto-layout?  
Example: /, /opt, /var  
If no, you must provide file system configuration information.                                                                                   |                           |
| Mount Remote           | Does this system need to access software on another file system?                                                                                                                                                   | Yes/No                    |
| File Systems (only available in the Solaris suninstall program) | If yes, provide the following information about the remote file system.  
Server:  
IP Address:  
Remote File System:  
Local Mount Point:                                                                 |                           |
Checklist for Upgrading

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

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

- If the system is part of a network, verify that an Ethernet connector or similar network adapter is connected to your system.

- If you are using the Solaris Web Start program from the Solaris 9 Installation CD, verify that you have a 512–Mbyte slice on the disk. For detailed information, including more requirements for x86 systems, refer to Table 2–4. You can ignore these requirements if you are installing from a DVD or an installation image.

- If you are using Solaris Live Upgrade, determine your resource requirements for creating a new boot environment and upgrading it. For detailed information, refer to Chapter 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 333.

- Verify that you have enough disk space. For more information, refer to Chapter 5.

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

- Review the Solaris 9 Release Notes and vendor release notes to ensure that the software you use is still supported in the new release.

- Review the Solaris 9 Sun Hardware Platform Guide to ensure that your hardware is supported.

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

- Review vendor and third-party software documentation for additional upgrade instructions.

- Check for all of the available patches that you might need. The most recent patch list is provided at http://sunsolve.sun.com.

- Check the system for the existence of Prestoserve software. If you begin the upgrade process by shutting down the system with the `init 0` command, you might lose data. Refer to the Prestoserve documentation for shutdown instructions.

- x86: If you are using the Linux operating system, the Solaris `fdisk` partition and the Linux `swap` partition use the same identifier, 0x82. To resolve the problem, you can do one of the following.
Choose not to use a Linux swap partition at all, provided that you have enough memory.

Put the Linux swap partition on another drive.

Back up the Linux data you want to keep to storage media, install the Solaris operating environment, and then reinstall Linux.

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

---

**Worksheet for Upgrading**

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

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

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

---

**TABLE 6–2 Upgrade Worksheet**

<table>
<thead>
<tr>
<th>Info Needed to Install</th>
<th>Description/Example</th>
<th>Enter Your Answers Here:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Is the system connected to a network?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>DHCP</td>
<td>Can the system use Dynamic Host Configuration Protocol (DHCP) to configure its network interfaces?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Host Name</td>
<td>Host name that you choose for the system.</td>
<td></td>
</tr>
</tbody>
</table>
TABLE 6–2 Upgrade Worksheet (Continued)

<table>
<thead>
<tr>
<th>Info Needed to Install</th>
<th>Description/Example</th>
<th>Enter Your Answers Here:</th>
</tr>
</thead>
</table>
| 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. |                         |
| NIS+ and NIS           | Do you want to specify a name server or let the installation program find one?  
If you want to specify a name server, provide the following information.  
Server’s host name:  
Server’s IP Address: | Specify One/Find One |
| DNS                    | Provide IP addresses for the DNS server. You must enter at least one IP address, but you can enter up to three addresses.  
Server’s IP Address(es):  
You can enter a list of domains to search when a DNS query is made.  
Search Domain: |                         |
<table>
<thead>
<tr>
<th>Info Needed to Install</th>
<th>Description/Example</th>
<th>Enter Your Answers Here:</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>Provide the following information about your LDAP profile.</td>
<td>Profile Name:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Profile Server:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP Address:</td>
</tr>
<tr>
<td></td>
<td>If you specify a proxy credential level in your LDAP profile, gather this information.</td>
<td>Proxy-bind distinguished name:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proxy-bind password:</td>
</tr>
<tr>
<td>Default Route</td>
<td>Do you want to specify a default route IP address or let the Solaris Web Start installation program or Solaris suninstall program find one?</td>
<td>Specify One/Find One/None</td>
</tr>
<tr>
<td></td>
<td>The default route provides a bridge that forwards traffic between two physical networks. An IP address is a unique number that identifies each host on a network.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You have the following choices:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ You can specify the IP address. An /etc/defaultrouter file is created with the specified IP address. When the system is rebooted, the specified IP address becomes the default route.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ You can let the software detect an IP address.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ The Solaris suninstall program detects an IP address when the system is rebooted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ The Solaris Web Start installation program can detect a default route. However, the system must be on a subnet that has a router that advertises itself by using the ICMP router discovery protocol.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ You can choose None if you do not have a router or do not want the software to detect an IP address at this time. The software automatically tries to detect an IP address on reboot.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Host IP address:</td>
<td></td>
</tr>
<tr>
<td>Time Zone</td>
<td>How do you want to specify your default time zone?</td>
<td>Geographic region</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Offset from GMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time zone file</td>
</tr>
<tr>
<td>Info Needed to Install</td>
<td>Description/Example</td>
<td>Enter Your Answers Here:</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Locales</td>
<td>For which geographic regions do you want to install support?</td>
<td></td>
</tr>
<tr>
<td>SPARC: Power Management (only available on SPARC systems that support Power Management)</td>
<td>Do you want to use Power Management?</td>
<td>Yes/No</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> – If your system has Energy Star version 3 or later, you are not prompted for this information.</td>
<td></td>
</tr>
<tr>
<td>Proxy Server Configuration (only available in the Solaris Web Start program)</td>
<td>Do you have a direct connection to the Internet or do you need to use a proxy server to gain access to the Internet?</td>
<td>Direct Connection/Proxy Server</td>
</tr>
<tr>
<td></td>
<td>If you use a proxy server, provide the following information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Host:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port:</td>
<td></td>
</tr>
<tr>
<td>Automatic reboot or CD/DVD ejection</td>
<td>Reboot automatically after software installation?</td>
<td>Yes/No</td>
</tr>
<tr>
<td></td>
<td>Eject CD/DVD automatically after software installation?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Disk Space Reallocation</td>
<td>Do you want the installation program to automatically re-lay out file systems on your disks?</td>
<td>Yes/No</td>
</tr>
<tr>
<td></td>
<td>If yes, which file systems should be used for auto-layout?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: <code>/</code>, <code>/opt</code>, <code>/var</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If no, you must provide information for file system configuration.</td>
<td></td>
</tr>
<tr>
<td>SPARC: 64–bit</td>
<td>Do you want to install support for 64–bit applications?</td>
<td>Yes/No</td>
</tr>
</tbody>
</table>
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 55
- “Ways to Preconfigure System Configuration Information” on page 56
- “Preconfiguring With the sysidcfg File” on page 57
- “Preconfiguring With the Name Service” on page 63
- “SPARC: Preconfiguring Power Management Information” on page 66

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

\(^1\) Yes if NIS or NIS+, No if DNS or LDAP
### TABLE 7-1 Methods to Preconfigure System Configuration Information (Continued)

<table>
<thead>
<tr>
<th>Preconfigurable System Information</th>
<th>Preconfigurable With the <code>sysidcfg</code> File?</th>
<th>Preconfigurable With the Name Service?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web proxy²</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>x86: Monitor type</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>x86: Keyboard language, keyboard layout</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>x86: Graphics card, color depth, display resolution, screen size</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>x86: Pointing device, number of buttons, IRQ level</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SPARC: Power Management (autoshutdown)³</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

1. Because this information is system specific, edit the name service rather than create a different `sysidcfg` file for each system.

2. You can configure this information during a Solaris Web Start installation, but not through the `sysidcfg` file or the name service.

3. You cannot preconfigure this system configuration information through the `sysidcfg` file or the name service. “SPARC: Preconfiguring Power Management Information” on page 66 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 or PCFS 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 217.

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:

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

<table>
<thead>
<tr>
<th>Syntax Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent keywords can be listed in any order.</td>
<td><code>pointer=MS-S</code></td>
</tr>
<tr>
<td></td>
<td><code>display=ati {size=15-inch}</code></td>
</tr>
<tr>
<td>Keywords are not case sensitive.</td>
<td><code>TIMEZONE=US/Central</code></td>
</tr>
<tr>
<td></td>
<td><code>terminal=PC Console</code></td>
</tr>
<tr>
<td>Enclose all dependent keywords in curly braces ()</td>
<td><code>name_service=NIS</code></td>
</tr>
</tbody>
</table>
| to tie them to their associated independent keyword. | `{domain_name=marquee.central.sun.com
name_server=connor(129.152.112.3)}`                                   |
| You can optionally enclosed values in single ('') | `network_interface='none'`                                              |
| or double quotes (").                           | `network_interface=none`                                                |
| 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. | `network_interface=le0`                                                  |
sysidcfg File Keywords

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

<table>
<thead>
<tr>
<th>Configuration Information</th>
<th>Keywords</th>
<th>Values or Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name service, domain name, name server</td>
<td>name_service=NIS, NIS+, DNS, LDAP, NONE</td>
<td>name_service=NIS {domain_name=domain_name, name_server=hostname(ip_address)}</td>
</tr>
</tbody>
</table>

Options for NIS and NIS+:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Options for NIS and NIS+:</td>
<td>name_service=NIS</td>
<td>{domain_name=domain_name, name_server=hostname(ip_address)}</td>
</tr>
<tr>
<td>Options for DNS:</td>
<td>name_service=DNS</td>
<td>{domain_name=domain_name, name_server=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)}</td>
</tr>
<tr>
<td>Options for LDAP:</td>
<td>name_service=LDAP</td>
<td>{domain_name=domain_name, profile=profile_name, profile_server=ip_address, proxy_dn=&quot;proxy_bind_dn&quot;, proxy_password=password}</td>
</tr>
</tbody>
</table>

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 {}.

Note – The proxy_dn and proxy_password keywords are optional. The proxy_dn value must be enclosed in double quotes.
### TABLE 7-2 Keywords You Can Use in `sysidcfg` (Continued)

<table>
<thead>
<tr>
<th>Configuration Information</th>
<th>Keywords</th>
<th>Values or Examples</th>
</tr>
</thead>
</table>
| Network interface, host name, Internet Protocol (IP) address, netmask, DHCP, IPv6 | network_interface=NONE, PRIMARY, or value | - 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
```

If DHCP is not to be used, specify:  
```
{hostname=host_name
default_route=ip_address or NONE
ip_address=ip_address
netmask=netmask
protocol_ipv6=yes_or_no}
```

**Note** – If you do not use DHCP, you are not required to specify protocol_ipv6=yes and default_route. But, take note of the following issues.  
- A JumpStart installation requires protocol_ipv6 to be specified now or you are prompted interactively later.  
- If default_route is not specified and one cannot be detected using the ICMP router discovery protocol, you are prompted to choose a route.

Root password | root_password=root_password | Encrypted from `/etc/shadow`.  
Security policy | security_policy=kerberos, NONE |  
Options for Kerberos:  
```
{default_realm=FQDN
admin_server=FQDN kdc=FQDN1, FQDN2, FQDN3}
```

**(FQDN is a fully qualified domain name)**  
**Note** – You can list a maximum of three key distribution centers (KDCs), but at least one is required.
### TABLE 7-2 Keywords You Can Use in `sysidcfg` (Continued)

<table>
<thead>
<tr>
<th>Configuration Information</th>
<th>Keywords</th>
<th>Values or Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language in which to display the install program and desktop</td>
<td><code>system_locale=locale</code></td>
<td>The <code>/usr/lib/locale</code> directory or Chapter 40 provides the valid locale values. The subdirectories in the <code>/usr/share/lib/terminfo</code> directory provide the valid terminal values. 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. 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.</td>
</tr>
<tr>
<td>Terminal type</td>
<td><code>terminal=terminal_type</code></td>
<td>On the system you want to install, run <code>kdmconfig -d filename</code>. Append output to <code>sysidcfg</code> file.</td>
</tr>
<tr>
<td>Time zone</td>
<td><code>timezone=timezone</code></td>
<td>On the system you want to install, run <code>kdmconfig -d filename</code>. Append output to <code>sysidcfg</code> file.</td>
</tr>
<tr>
<td>Date and time</td>
<td><code>timeserver=localhost, hostname, ip_addr</code></td>
<td>On the system you want to install, run <code>kdmconfig -d filename</code>. Append output to <code>sysidcfg</code> file.</td>
</tr>
<tr>
<td>x86: Monitor type</td>
<td><code>monitor=monitor_type</code></td>
<td>On the system you want to install, run <code>kdmconfig -d filename</code>. Append output to <code>sysidcfg</code> file.</td>
</tr>
<tr>
<td>x86: Keyboard language, keyboard layout</td>
<td><code>keyboard=keyboard_language</code> <code>{layout=value}</code></td>
<td>On the system you want to install, run <code>kdmconfig -d filename</code>. Append output to <code>sysidcfg</code> file.</td>
</tr>
<tr>
<td>x86: Graphics card, screen size, color depth, display resolution</td>
<td><code>display=graphics_card</code> <code>{size=size depth=color_depth resolution=screen_resolution}</code></td>
<td>On the system you want to install, run <code>kdmconfig -d filename</code>. Append output to <code>sysidcfg</code> file.</td>
</tr>
<tr>
<td>x86: Pointing device, number of buttons, IRQ level</td>
<td><code>pointer=pointing_device</code> <code>{nbuttons=number_buttons irq=value}</code></td>
<td>On the system you want to install, run <code>kdmconfig -d filename</code>. Append output to <code>sysidcfg</code> file.</td>
</tr>
</tbody>
</table>

▼ 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 or PCFS diskette.

**SPARC: Example `sysidcfg` File**

The following is an example of a `sysidcfg` file for a group of SPARC based systems. 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.

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

**x86: Example `sysidcfg` File**

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

```plaintext
keyboard=ATKB {layout=US-English}
display=ati {size=15-inch}
pointer=MS-S
```
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.

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

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 63
- “To Preconfigure the Locale Using NIS+” on page 65

▼ To Preconfigure the Locale Using NIS

1. Become superuser on the name server.

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

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

   ```
   locale.time: $(DIR)/locale
   -@if [ -f $(DIR)/locale ]; then \n     sed -e "/\#d" -e s/\.\*$/// $(DIR)/locale \n     | awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \n     | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/localebyname; \n   ```
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'
```


name

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

The locale you want to install on the system and use on the desktop after the system is rebooted. Chapter 40 contains a list of valid locales.

comment

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

Note — Locales are available on the Solaris 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.

SPARC: Preconfiguring Power Management Information

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

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

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

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

touch /a/autoshutdown
Finish scripts are described in “Creating Finish Scripts” on page 237.
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 69
- “Using Solaris Live Upgrade” on page 71
- “Using Custom JumpStart to Upgrade” on page 71
- “Upgrading With Disk Space Reallocation” on page 72
- “Backing Up Systems Before Upgrading” on page 73

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

Note – Use the `smosservice patch` to upgrade diskless clients. For detailed instructions, refer to `System Administration Guide: Basic Administration` or to `smosservice(1M)`. 
If you are already running the Solaris 9 operating environment and have installed individual patches, upgrading to a Solaris 9 Update release causes the following:

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

You can use the Patch Analyzer to determine which patches, if any, will be removed by upgrading to the Solaris 9 Update release. For detailed instructions about using the Patch Analyzer, refer to “Upgrading to a Solaris Update Release” on page 471.

### Solaris Upgrade Methods

**TABLE 8-1** SPARC: Solaris Upgrade Methods

<table>
<thead>
<tr>
<th>Platform</th>
<th>Current Solaris Operating Environment</th>
<th>Solaris Upgrade Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPARC systems</td>
<td>Solaris 2.6, Solaris 7, Solaris 8, Solaris 9</td>
<td>- The Solaris Web Start program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Solaris suninstall program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Custom JumpStart method</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Solaris Live Upgrade</td>
</tr>
</tbody>
</table>

**TABLE 8-2** x86: Solaris Upgrade Methods

<table>
<thead>
<tr>
<th>Current Solaris Operating Environment</th>
<th>Solaris Upgrade Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 2.6</td>
<td>Installing from DVD media or a net installation image:</td>
</tr>
<tr>
<td></td>
<td>- Solaris Web Start program</td>
</tr>
<tr>
<td></td>
<td>- Solaris suninstall program</td>
</tr>
<tr>
<td></td>
<td>- Custom JumpStart method</td>
</tr>
<tr>
<td></td>
<td>Installing from CD media:</td>
</tr>
<tr>
<td></td>
<td>- Solaris suninstall program</td>
</tr>
<tr>
<td></td>
<td>- Custom JumpStart method</td>
</tr>
<tr>
<td>Solaris 7</td>
<td>Installing from DVD media or a net installation image:</td>
</tr>
<tr>
<td></td>
<td>- Solaris Web Start program</td>
</tr>
<tr>
<td></td>
<td>- Solaris suninstall program</td>
</tr>
<tr>
<td></td>
<td>- Custom JumpStart method</td>
</tr>
<tr>
<td></td>
<td>- Solaris Live Upgrade</td>
</tr>
<tr>
<td></td>
<td>Installing from CD media:</td>
</tr>
<tr>
<td></td>
<td>- Solaris suninstall program</td>
</tr>
<tr>
<td></td>
<td>- Custom JumpStart method</td>
</tr>
<tr>
<td></td>
<td>- Solaris Live Upgrade</td>
</tr>
</tbody>
</table>
TABLE 8-2 x86: Solaris Upgrade Methods  (Continued)

<table>
<thead>
<tr>
<th>Current Solaris Operating Environment</th>
<th>Solaris Upgrade Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 8, Solaris 9</td>
<td>Installing from DVD or CD media or a net installation image:</td>
</tr>
<tr>
<td></td>
<td>- Solaris Web Start program</td>
</tr>
<tr>
<td></td>
<td>- Solaris suninstall program</td>
</tr>
<tr>
<td></td>
<td>- Custom JumpStart method</td>
</tr>
<tr>
<td></td>
<td>- Solaris Live Upgrade</td>
</tr>
</tbody>
</table>

**Note** – For limitations on upgrading using Solaris Live Upgrade, see “Solaris Live Upgrade System Requirements” on page 329.

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

### Using Solaris Live Upgrade

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

For instructions on how to plan for and use Solaris Live Upgrade, see Chapter 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 229.

You can use JumpStart to update a clone system with a Solaris Flash differential archive. For an overview of Solaris Flash archives, see Chapter 17.
Upgrading With Disk Space Reallocation

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

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

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

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

- If you are using the Solaris Web Start program, and auto-layout cannot determine how to reallocate the disk space, you must use the Solaris `suninstall` program or the custom JumpStart program to upgrade.
- If you are using the Solaris `suninstall` program, and auto-layout cannot determine how to reallocate disk space, you must specify the file systems that can be moved or changed and run auto-layout again.
- If you are using the custom JumpStart method to upgrade and you create an upgrade profile, disk space might be a concern. If the current file systems do not contain enough disk space for the upgrade, you can use the `backup_media` and `layout_constraint` keywords to reallocate disk space. For an example of how to use the `backup_media` and `layout_constraint` keywords in a profile, refer to Example 23–5.
Backing Up Systems Before Upgrading

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

<table>
<thead>
<tr>
<th>Chapter 10</th>
<th>Provides overview and planning information for installing Solaris software from an install server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 11</td>
<td>Provides step-by-step instructions for copying Solaris software from DVD media to an install server.</td>
</tr>
<tr>
<td>Chapter 12</td>
<td>Provides step-by-step instructions for copying Solaris software from CD media to an install server.</td>
</tr>
<tr>
<td>Chapter 13</td>
<td>Describes commands to set up network installations.</td>
</tr>
</tbody>
</table>
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 CD

After 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 and for multiple platforms by copying the images on to the install server’s hard disk. For example, a single install server could contain the disc images for the SPARC platform and x86 platform.

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

- “To Create a SPARC Install Server With SPARC or x86 DVD Media” on page 83
- “x86: To Create an x86 Install Server With SPARC or x86 DVD Media” on page 86
- “SPARC: To Create a SPARC Install Server With SPARC CD Media” on page 97
- “Creating a Cross-Platform Install Server for CD Media” on page 106

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

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

  **Note** – When using DHCP, you do not need to create a separate boot server. For more information, see “Using DHCP to Provide Network Installation Parameters” on page 79

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 89
- “Creating a Boot Server on a Subnet With a CD Image” on page 117

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

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

  **Note** – The install server and name server can be the same or different systems.
Figure 10–1 illustrates the servers that are typically used for network installation.

Using DHCP to Provide Network Installation Parameters

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

For information on DHCP options for installation parameters, see “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 81
- “Creating an Install Server With DVD Media” on page 82
- “Creating a Boot Server on a Subnet With a DVD Image” on page 89
- “Adding Systems to Be Installed From the Network With a DVD Image” on page 91

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an install server.</td>
<td>Use the <code>setup_install_server(1M)</code> command to copy the Solaris 9 DVD to the install server’s hard disk.</td>
<td>“Creating an Install Server With DVD Media” on page 82</td>
</tr>
</tbody>
</table>
TABLE 11–1 Task Map: Setting Up an Install Server With DVD Media  (Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) Create boot servers.</td>
<td>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.</td>
<td>“Creating a Boot Server on a Subnet With a DVD Image” on page 89</td>
</tr>
<tr>
<td>Add systems to be installed from the network.</td>
<td>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.</td>
<td>“Adding Systems to Be Installed From the Network With a DVD Image” on page 91</td>
</tr>
</tbody>
</table>

Creating an Install Server With DVD Media

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

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

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

- Solaris 2.6 *SPARC Platform Edition* operating environment — Patch ID 107618-03
- Solaris 2.6 *Intel Platform Edition* operating environment — Patch ID 107619-03
- Solaris 7 *SPARC Platform Edition* operating environment — Patch ID 107259-03
- Solaris 7 *Intel Platform Edition* operating environment — Patch ID 107260-03
To Create a SPARC Install Server With SPARC or x86 DVD Media

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

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

1. On the SPARC system that is to become the install server, become superuser.
   The system must include a DVD-ROM drive and be part of the site’s network and name service. If you use a name service, the system must already be in a service, such as NIS, NIS+, DNS, or LDAP. If you do not use a name service, you must distribute information about this system by following your site’s policies.

2. Insert the Solaris 9 DVD in the SPARC system’s drive.

3. Create a directory to contain the DVD image.
   
   ```
   # mkdir -p install_dir_path
   ```

4. Change to the Tools directory on the mounted disc.
   
   ■ For SPARC DVD media, type:
   
   ```
   # cd /cdrom/cdrom0/s0/Solaris_9/Tools
   ```
   
   ■ For x86 DVD media, type:
   
   ```
   # cd /cdrom/cdrom0/Solaris_9/Tools
   ```
   
   In the previous examples, cdrom0 is the path to the drive that contains the Solaris operating environment DVD media.

5. Copy the DVD image in the drive to the install server’s hard disk.
   
   ```
   # ./setup_install_server install_dir_path
   ```

   *install_dir_path* Specifies the directory where the DVD image is to be copied.
**Note** – The `setup_install_server` command indicates whether you have enough disk space available for the Solaris 9 Software disc images. To determine available disk space, use the `df -lk` command.

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

7. Verify that the path to the install server’s image is shared appropriately.
   ```
   # share | grep install_dir_path
   ```
   `install_dir_path` specifies the path to the installation image where the DVD image was copied
   - If the path to the install server’s directory is displayed and `anon=0` is displayed in the options, proceed to Step 10.
   - If the path to the install server’s directory is not displayed or you do not have `anon=0` in the options, continue.

8. Make the install server available to the boot server by adding this entry to the `/etc/dfs/dfstab` file.
   ```
   share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
   ```

9. Make sure that the install server’s directory path is correctly shared.
   ```
   # ps -ef | grep nfss
   ```
   - If the `nfss` daemon is not running, start it.
     ```
     # /etc/init.d/nfs.server start
     ```
   - If the `nfss` daemon is running, share the install server.
     ```
     # shareall
     ```

10. Change directories to root (`/`).
    ```
    # cd /
    ```

11. Eject the Solaris 9 DVD.

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

- If you are using DHCP or the install server is on the same subnet as the system to be installed, you do not need to create a boot server. Proceed to “Adding Systems to Be Installed From the Network With a DVD Image” on page 91.
- 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 89.

**EXAMPLE 11–1 SPARC: Creating a SPARC Install Server With a SPARC DVD**

The following example illustrates how to create an install server by copying the Solaris 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:

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/s9dvdsparc
```

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

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

**EXAMPLE 11–2 x86: Creating a SPARC Install Server With an x86 DVD**

The following example illustrates how to create an install server by copying the Solaris 9 DVD to the install server’s `/export/home/s9dvdx86` directory:

```
# mkdir -p /export/home/s9dvdx86
# cd /cdrom/cdrom0/Solaris_9/Tools
#. /setup_install_server /export/home/s9dvdx86
```

Add the following path to the `/etc/dfs/dfstab` file:
EXAMPLE 11-2  x86: Creating a SPARC Install Server With an x86 DVD  
(Continued)

share -F nfs -o ro,anon=0 -d "install server directory" /export/home/s9dvdx86

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

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

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

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

1. On the x86 system that is to become the install server, become superuser.
   The system must include a DVD-ROM drive and be part of the site’s network and name service. If you use a name service, the system must also be in the NIS, NIS+, DNS, or LDAP name service. If you do not use a name service, you must distribute information about this system by following your site’s policies.

2. Insert the Solaris 9 DVD into the system’s drive.

3. Create a directory to contain the boot image.
   # mkdir -p install_dir_path

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

4. Change to the Tools directory on the mounted disc:
   ■ For x86 DVD media, type:
     # cd /cdrom/cdrom0/s2/Solaris_9/Tools
   ■ For SPARC DVD media, type:
     # cd /cdrom/cdrom0/Solaris_9/Tools

   In the previous examples, cdrom0 is the path to the drive that contains the Solaris operating environment DVD media.
5. Copy the disc in the drive to the install server’s hard disk by using the `setup_install_server` command:

```bash
# ./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 -k` command.

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

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

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

```bash
# share | grep install_dir_path
```

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

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

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

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

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

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

- If the `nfsd` daemon is not running, start it.
  ```bash
  # /etc/init.d/nfs.server start
  ```
- If the `nfsd` daemon is running, share the install server.
  ```bash
  # shareall
  ```

10. Change directories to root (`/`).
11. Eject the Solaris 9 DVD.

12. Decide if you want to patch the files that are located in the miniroot (Solaris_9/Tools/Boot) on the net install image that was created by setup_install_server.
   - If no, continue.
   - If yes, use the patchadd -C command to patch the files that are located in the miniroot.

13. Decide if you need to create a boot server.
   - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. See “Adding Systems to Be Installed From the Network With a DVD Image” on page 91.
   - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must create a boot server. For detailed instructions on how to create a boot server, refer to “Creating a Boot Server on a Subnet With a DVD Image” on page 89.

Example 11-3 x86: Creating an x86 Install Server With an x86 DVD

The following example illustrates how to create an x86 install server by copying the Solaris 9 x86 Platform Edition DVD to the install server’s /export/home/s9dvdx86 directory:

```
# mkdir -p /export/home/s9dvdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/s9dvdx86
```

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

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/s9dvdx86
```

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

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

Example 11-4 Creating an x86 Install Server With a SPARC DVD

The following example illustrates how to create an x86 install server by copying the Solaris 9 SPARC Platform Edition DVD to the install server’s /export/home/s9dvdsparc directory:

```
# mkdir -p /export/home/s9dvdsparc
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./setup_install_server /export/home/s9dvdsparc
```
EXAMPLE 11–4 Creating an x86 Install Server With a SPARC DVD  (Continued)

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

```
share  -F nfs  -o ro,anon=0 -d "install server directory" /export/home/s9dvdsparc
```

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

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

Creating a Boot Server on a Subnet With a DVD Image

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

- If you are using DHCP to set installation parameters or your install server or client 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 With a DVD Image” on page 91.
- 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

\textit{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

\textit{boot_dir_path} Specifies the directory where the boot software is to be copied

4. Change to the \texttt{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

\texttt{-b} Specifies to set up the system as a boot server

\textit{boot_dir_path} Specifies the directory where the boot software is to be copied

\begin{quote}
\textbf{Note} – The \texttt{setup_install_server} command indicates whether you have enough disk space available for the images. To determine available disk space, use the \texttt{df -kl} command.
\end{quote}

6. Change directories to root (/).

\# cd /

7. Unmount the installation image.

\# umount /mnt

You are now ready to set up systems to be installed from the network. See “Adding Systems to Be Installed From the Network With a DVD Image” on page 91.

\textbf{EXAMPLE 11-5} Creating a Boot Server on a Subnet (DVD)

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

\# mount -F nfs -o ro crystal:/export/home/s9dvdsparc /mnt
\# mkdir -p /export/home/s9dvdsparc
\# cd /mnt/Solaris_9/Tools
Adding Systems to Be Installed From the Network With a DVD Image

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

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

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

- If you are using DHCP to set installation parameters, see Example 11–6
- If your install server and client are on the same subnet, see Example 11–7
- If your install server and your client are not on the same subnet and you are not using DHCP, see Example 11–8

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 “To Create a SPARC Install Server With SPARC or x86 DVD Media” Step 6.

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

- Host name
- IP address
- Ethernet address

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

3. Change to the **Tools** directory on the Solaris 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
```

- `-d` Specifies that the client is to use DHCP to obtain the network install parameters.
  
  For x86 clients, use this option to boot the systems from the network by using PXE network boot.

- `-s install_server:install_dir_path` Specifies the name and path to the install server.

  - `install_server` is the host name of the install server
  - `install_dir_path` is the absolute path to the Solaris 9 DVD image

- `-c jumpstart_server:jumpstart_dir_path` Specifies a JumpStart directory for custom JumpStart installations. `jumpstart_server` is the host name of the server on which the JumpStart directory is located. `jumpstart_dir_path` is the absolute path to the JumpStart directory.

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

- `-t boot_image_path` Specifies the path to an alternate boot image if you want to use a boot image other than the one in the **Tools** directory on the Solaris 9 net installation image, CD, or DVD.
**client_name**
Is the name of the system to be installed from the network. This name is *not* the host name of the install server.

**platform_group**
Is the platform group of the system to be installed. For more information, see Chapter 39.

**EXAMPLE 11-6** SPARC: Adding a SPARC Install Client on a SPARC Install Server When Using DHCP (DVD)

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

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

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

**EXAMPLE 11-7** Adding an Install Client That Is On the Same Subnet As Its Server (DVD)

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

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

**EXAMPLE 11-8** Adding an Install Client to a Boot Server (DVD)

The following example illustrates how to add an install client to a boot server. The install client is named *rose*, which is an Ultra™ 5 system. Run the command on the boot server. The `-s` option is used to specify an install server that is named *rosemary*, which contains a Solaris 9 SPARC Platform Edition 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
```

**EXAMPLE 11-9** x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP (DVD)

The following example illustrates how to add an x86 install client to an install server when you are using DHCP to set installation parameters on the network. The `-d` option is used to specify that clients are to use the DHCP protocol for configuration. If you plan to use PXE network boot, you must use the DHCP protocol. The DHCP class
EXAMPLE 11-9 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP (DVD) (Continued)

name SUNW.i86pc indicates that this command applies to all Solaris x86 network boot clients, not just a single client. The -s option is used to specify that the clients are to be installed from the install server that is named rosemary. This server contains a Solaris 9 x86 Platform Edition DVD image in /export/boot/s9dvdx86.

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

```
x86_install_server# cd /export/boot/s9dvdx86/Solaris_9/Tools
x86_install_server# ./add_install_client -d -s rosemary:/export/home/s9dvdx86 \  
SUNW.i86pc i86pc
```
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 96
- “Creating a SPARC Install Server With CD Media” on page 96
- “Creating an x86 Install Server With CD Media” on page 101
- “Creating a Cross-Platform Install Server for CD Media” on page 106
- “Creating a Boot Server on a Subnet With a CD Image” on page 117
- “Adding Systems to Be Installed From the Network With a CD Image” on page 119
Task Map: Preparing to Install From the Network With CD Media

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an install server.</td>
<td>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.</td>
<td>“Creating a SPARC Install Server With CD Media” on page 96 “x86: To Create an x86 Install Server With x86 CD Media” on page 102 “Creating a Cross-Platform Install Server for CD Media” on page 106</td>
</tr>
<tr>
<td>(Optional) Create boot servers.</td>
<td>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.</td>
<td>“Creating a Boot Server on a Subnet With a CD Image” on page 117</td>
</tr>
<tr>
<td>Add systems to be installed from the network.</td>
<td>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.</td>
<td>“Adding Systems to Be Installed From the Network With a CD Image” on page 119</td>
</tr>
</tbody>
</table>

Creating a SPARC Install Server With CD Media

The install server contains the installation image needed to install systems from the network. You must create an install server to install the Solaris software on a system from the network. You do not always need to set up a separate boot server.
- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a separate boot server.
- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.

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

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

If you want to create an install server by using media of a platform different from the install server, for example, a SPARC system with x86 CD media, see “Creating a Cross-Platform Install Server for CD Media” on page 106

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

5. **Copy the image in the drive to the install server’s hard disk.**
   
   ```
   # ./setup_install_server install_dir_path
   ```
install_dir_path

Specifies the directory where the CD image is to be copied

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

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

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

   # share | grep install_dir_path

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

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

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

   
   install_dir_path
   
   Specifies the path to the boot server installation image

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

   # ps -ef | grep nfsd
   
   - If the nfsd daemon is not running, start it.
     
     # /etc/init.d/nfs.server start
   
   - If the nfsd daemon is running, share the install server.
     
     # shareall

10. Change directories to root (/).

    # cd /
11. Eject the Solaris 9 Software 1 of 2 CD.

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

13. Change to the Tools directory on the mounted CD.
   
   ```
   # cd /cdrom/cdrom0/Solaris_9/Tools
   ```

14. Copy the CD in the CD-ROM drive to the install server's hard disk.
   
   ```
   # ./add_to_install_server install_dir_path
   ```

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

15. Change directories to root (/).
   
   ```
   # cd /
   ```

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

17. Insert the Solaris 9 Languages CD in the system’s CD-ROM drive.

18. Change to the Tools directory on the mounted CD.
   
   ```
   # cd /cdrom/cdrom0/Tools
   ```

19. Copy the CD in the CD-ROM drive to the install server's hard disk.
   
   ```
   # ./add_to_install_server install_dir_path
   ```

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

20. Change directories to root (/).
   
   ```
   # cd /
   ```

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

   - If no, eject the Solaris 9 Languages CD and proceed to Step 25.
   - If yes, eject the Solaris 9 Languages CD and continue.

22. Insert the Solaris 9 Installation CD into the system’s CD-ROM drive.

23. Change to the directory that contains `modify_install_server` on the mounted CD:
   
   ```
   # cd /cdrom/cdrom0/s0
   ```

24. Copy the Solaris Web Start interface software to the install server.
# ./modify_install_server -p install_dir_path installer_miniroot_path

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

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

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

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

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

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

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

- If you are using DHCP or the install server is on the same subnet as the system to be installed, you do not need to create a boot server. Proceed to “Adding Systems to Be Installed From the Network With a CD Image” on page 119.
- 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 CD Image” on page 117.

**EXAMPLE 12-1 SPARC: Creating a SPARC Install Server With SPARC CD Media**

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

- Solaris 9 Software 1 of 2 SPARC Platform Edition CD
- Solaris 9 Software 2 of 2 SPARC Platform Edition CD
- Solaris 9 SPARC Platform Edition 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:
EXAMPLE 12-1 SPARC: Creating a SPARC Install Server With SPARC CD Media
(Continued)

share -F nfs -o ro,anon=0 -d "install server directory" \ 
/export/home/s9cdsparc

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

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

Continue with the following steps.

- If you do not need a boot server or have completed the steps for a separate boot server, continue.

# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./add_to_install_server /export/home/s9cdsparc
# cd /
# cd /cdrom/cdrom0/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 an x86 Install Server With CD Media

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

- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a separate boot server.

- If your install server and your client are not on the same subnet and you are not using DHCP, you must create separate boot servers for each subnet. You could create an install server for each subnet; however, install servers require more disk space.
x86: To Create an x86 Install Server With x86 CD Media

This procedure creates an x86 install server with x86 CD media.

If you want to create an install server by using media of a platform different from the install server, for example, an x86 system with SPARC CD media, see “Creating a Cross-Platform Install Server for CD Media” on page 106.

**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/s2/Solaris_9/Tools
   ```
   In the previous example, *cdrom0* is the path to the drive that contains the Solaris operating environment CD media.

5. Copy the image in the drive to the install server’s hard disk.
   ```
   # ./setup_install_server install_dir_path
   ```
   
   *install_dir_path* Specifies the directory where the CD image is to be copied
6. Decide if you need to make the install server available for mounting.
   - If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. Proceed to Step 10.
   - If the install server is not on the same subnet as the system to be installed and you are not using DHCP, complete the following steps.

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

   ```
   # share | grep install_dir_path
   ```

   `install_dir_path` Specifies the path to the boot-server installation image

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

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

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

   `install_dir_path` Specifies the path to the boot-server installation image

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

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

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

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

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

     ```
     # shareall
     ```

10. Change directories to root (/).

    ```
    # cd /
    ```

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

12. Insert the Solaris 9 Software 2 of 2 CD in the system's CD-ROM drive.
13. Change to the Tools directory on the mounted CD:
   
   ```bash
   # cd /cdrom/cdrom0/Solaris_9/Tools
   ```

14. Copy the CD in the CD-ROM drive to the install server’s hard disk.
   
   ```bash
   # ./add_to_install_server install_dir_path
   ```

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

15. Change directories to root (/).
   
   ```bash
   # cd /
   ```

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

17. Insert the Solaris 9 Languages CD in the system’s CD-ROM drive.

18. Change to the Tools directory on the mounted CD:
   
   ```bash
   # cd /cdrom/cdrom0/Tools
   ```

19. Copy the CD in the CD-ROM drive to the install server’s hard disk.
   
   ```bash
   # ./add_to_install_server install_dir_path
   ```

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

20. Change directories to root (/).
   
   ```bash
   # cd /
   ```

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

   - If no, eject the Solaris 9 Languages CD and proceed to Step 25.
   - If yes, eject the Solaris 9 Languages CD and continue.

22. Insert the Solaris 9 Installation CD into the system’s CD-ROM drive.

23. Change to the directory that contains modify_install_server on the mounted CD:
   
   ```bash
   # cd /cdrom/cdrom0/s2
   ```

24. Copy the Solaris Web Start interface software to the install server.
   
   ```bash
   # ./modify_install_server -p install_dir_path installer_miniroot_path
   ```
25. Decide if you want to patch the files that are located in the miniroot
(/install_dir_path/Solaris_9/Tools/Boot) on the net install image that was
created by setup_install_server. Patching a file might be necessary if a boot
image has problems.
   ■ If no, continue.
   ■ If yes, use the patchadd -C command to patch the files that are located in the
     miniroot.

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

26. Decide if you need to create a boot server.
   ■ If you are using DHCP or the install server is on the same subnet as the system to
     be installed, you do not need to create a boot server. Proceed to “Adding Systems
to Be Installed From the Network With a CD Image” on page 119.
   ■ 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 CD Image” on page 117.

EXAMPLE 12-2 x86: Creating an x86 Install Server With x86 CD Media

The following example illustrates how to create an install server by copying the
following CDs to the install server’s /export/home/s9cdx86 directory:

- Solaris 9 Software 1 of 2 x86 Platform Edition CD
- Solaris 9 Software 2 of 2 x86 Platform Edition CD
- Solaris 9 x86 Platform Edition Languages CD

# mkdir -p /export/home/s9cdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/s9cdx86

■ 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" \\n/export/home/s9cdx86
EXAMPLE 12-2 x86: Creating an x86 Install Server With x86 CD Media  (Continued)

Check if the \texttt{nfsd} daemon is running. If the \texttt{nfsd} daemon is not running, start it and share it.

\begin{verbatim}
# ps -ef | grep nfsd
#/etc/init.d/nfs.server start
# shareall
\end{verbatim}

Continue with the following steps.

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

\begin{verbatim}
# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
#/ .add_to_install_server /export/home/s9cdx86
# cd /
# cd /cdrom/cdrom0/Tools
#/ .add_to_install_server /export/home/s9cdx86
# cd /
# cd /cdrom/cdrom0/s2
#/ .modify_install_server -p /export/home/s9cdx86 /cdrom/cdrom0/s2
\end{verbatim}

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

---

Creating a Cross-Platform Install Server for CD Media

If you need to use a CD of a platform different from the install server, you cannot read the CD in the install server. You need a remote system to read the CD. For example, if you are setting up a SPARC install server and need to use x86 CD media, you need a remote x86 system to read the CDs.

\begin{itemize}
    \item To Create an x86 Install Server on a SPARC System With x86 CD Media
\end{itemize}

Use this procedure to create an x86 install server on a SPARC system with x86 CD media. You need the following:

- A SPARC system
- An x86 system with a CD-ROM drive
A set of CDs for the remote x86 system

- Solaris 9 Software 1 of 2 x86 Platform Edition CD
- Solaris 9 Software 2 of 2 x86 Platform Edition CD
- Solaris 9 x86 Platform Edition Languages CD
- Solaris 9 x86 Platform Edition Installation CD

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

In this procedure, *SPARC system* is the SPARC system that is to be the install server and *remote-x86-system* is the remote x86 system to be used with the x86 CD media.

1. **On the remote x86 system, become superuser.**
   
   The system must include a CD-ROM drive and be part of the site's network and name service. If you use a name service, the system must also be in the NIS, NIS+, DNS, or LDAP name service. If you do not use a name service, you must distribute information about this system by following your site's policies.

2. **On the remote x86 system, insert the Solaris 9 Software 1 of 2 x86 Platform Edition CD into the system's drive.**

3. **On the remote x86 system, add the following entries to the **/etc/dfs/dfstab** file.**

   ```
   share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
   share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
   ```

4. **On the remote x86 system, start the NFS daemon.**

   ```
   remote-x86-system# /etc/init.d/nfs.server start
   ```

5. **On the remote x86 system, verify that the CD is available to other systems by using the share command.**

   ```
   remote-x86-system# share
   - /cdrom/sol_9_x86/s0 ro,anon=0 " "
   - /cdrom/sol_9_x86/s2 ro,anon=0 " "
   ```

   In the previous sample output, **sol_9_x86** refers to the Solaris 9 (x86 Platform Edition) operating environment. This text string varies for each version of the Solaris operating environment.

6. **On the SPARC system that is to be the x86 install server, become superuser.**

7. **On the SPARC system, access the x86 CD by creating two directories for the appropriate mount points, one for the miniroot and one for the product.**

   ```
   SPARC-system# mkdir directory_name_s0
   ```
8. Verify that the CD is properly exported on the remote x86 system.

   SPARC-system# showmount -e remote-x86-system
   export list for remote-x86-system:
   /cdrom/sol_9_x86/s0 (everyone)
   /cdrom/sol_9_x86/s2 (everyone)

9. On the SPARC system, mount the remote x86 CD image.

   SPARC-system# mount remote_x86_system:/cdrom/sol_9_x86/s0 directory_name_s0
   SPARC-system# mount remote_x86_system:/cdrom/sol_9_x86/s2 directory_name_s2

10. On the SPARC system, change to the Tools directory on the mounted disc:

    SPARC-system# cd /directory_name_s2/Solaris_9/Tools

11. On the SPARC system, copy the disc in the drive to the install server's hard disk in
    the directory you've created by using the setup_install_server command:

    SPARC-system# ./setup_install_server -t directory_name_s0 install_dir_path

    -t                   Specifies the path to a boot image if you want to use a boot image
                         other than the one in the Tools directory on the Solaris 9 Software 2
                         of 2 CD.

    directory_name_s0   Is the name of the directory that contains the miniroot from slice 0.

    install_dir_path    Specifies the directory where the disc image is to be copied. The
                         directory must be empty.

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

12. On the SPARC system, change to the top directory.

    SPARC-system# cd /

13. On the SPARC system, unmount both directories.

    SPARC-system# unmount directory_name_s0
14. On the x86 system, unshare both CD-ROM slices.

\[ \text{remote x86 system} \# \text{unshare} /cdrom/sol_9\_x86/s0 \]

\[ \text{remote x86 system} \# \text{unshare} /cdrom/sol_9\_x86/s2 \]

15. On the x86 system, eject the Solaris 9 Software 1 of 2 x86 Platform Edition CD.

16. Insert the Solaris 9 Software 2 of 2 x86 Platform Edition CD into the SPARC system’s CD-ROM drive.

17. On the SPARC system, change to the Tools directory on the mounted CD:

\[ \text{SPARC-system} \# \text{cd} /cdrom/cdrom0/Solaris\_9/Tools \]

18. On the SPARC system, copy the CD to the install server’s hard disk:

\[ \text{SPARC-system} \# ./add\_to\_install\_server \text{install\_dir\_path} \]

install\_dir\_path Specifies the directory where the CD image is to be copied

19. Eject the Solaris 9 Software 2 of 2 x86 Platform Edition CD.

20. On the SPARC system, insert the Solaris 9 x86 Platform Edition Languages CD into the SPARC system’s CD-ROM drive and mount the CD.

21. On the SPARC system, change to the Tools directory on the mounted CD:

\[ \text{SPARC-system} \# \text{cd} /cdrom/cdrom0/Tools \]

22. On the SPARC system, copy the CD to the install server’s hard disk:

\[ \text{SPARC-system} \# ./add\_to\_install\_server \text{install\_dir\_path} \]

install\_dir\_path Specifies the directory where the CD image is to be copied

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

- If no, eject the Solaris 9 x86 Platform Edition Languages CD and proceed to Step 31
- If yes, eject the Solaris 9 x86 Platform Edition Languages CD and continue.

24. On the remote x86 system, insert the Solaris 9 x86 Platform Edition Installation CD into the x86 system’s CD-ROM drive.

25. On the remote x86 system, add the following entry to the /etc/dfs/dfstab file.

\[ \text{share} -F \text{nfs} -o \text{ro},\text{anon}=0 /cdrom/cdrom0/s0 \]

\[ \text{share} -F \text{nfs} -o \text{ro},\text{anon}=0 /cdrom/cdrom0/s2 \]
26. On the remote x86 system, start the NFS daemon.

```
remote-x86-system# /etc/init.d/nfs.server start
```

27. On the remote x86 system, verify that the CD is available to other systems:

```
remote-x86-system# share
  - /cdrom/multi_sol_9_x86/s0 ro,anon=0 " "
  - /cdrom/multi_sol_9_x86/s2 ro,anon=0 " 
```

28. On the SPARC system, mount the slices on the CD.

```
SPARC-system# mount remote_x86_system_name:/cdrom/multi_sol_9_x86/s0 directory_name_s0
SPARC-system# mount remote_x86_system_name:/cdrom/multi_sol_9_x86/s2 directory_name_s2
```

29. On the SPARC system, change to the directory that contains `modify_install_server` on the mounted CD:

```
SPARC-system# cd directory_name_s2
```

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

```
SPARC-system# ./modify_install_server -p install_dir_path directory_name_s0
```

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

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

`directory_name_s0` Name of the directory to contain the miniroot from slice 0

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

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

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

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

- If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. See “Adding Systems to Be Installed From the Network With a CD Image” on page 119
If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must create a boot server. For detailed instructions on how to create a boot server, refer to “To Create a Boot Server on a Subnet With a CD Image” on page 118.

EXAMPLE 12-3 Creating an x86 Install Server on a SPARC System With x86 CD Media

The following example illustrates how to create an x86 install server on a SPARC system that is named rosemary. The following x86 CDs are copied from a remote x86 system that is named tadpole to the SPARC install server’s /export/home/s9cdx86 directory.

- Solaris 9 Software 1 of 2 x86 Platform Edition CD
- Solaris 9 Software 2 of 2 x86 Platform Edition CD
- Solaris 9 x86 Platform Edition Languages CD
- Solaris 9 Installation Multilingual x86 Platform Edition CD

On the remote x86 system:

```bash
tadpole (remote-x86-system) # share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
```

```bash
tadpole (remote-x86-system) # share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
```

```bash
tadpole (remote-x86-system) # /etc/init.d/nfs.server start
```

On the SPARC system:

```bash
rosemary (SPARC-system) # mkdir /x86S0
```

```bash
rosemary (SPARC-system) # mkdir /x86S2
```

```bash
rosemary (SPARC-system) # mount tadpole:/cdrom/sol_9_x86/s0 /x86S0
```

```bash
rosemary (SPARC-system) # mount tadpole:/cdrom/sol_9_x86/s0 /x86S2
```

```bash
rosemary (SPARC-system) # cd /x86S2/Solaris_9/Tools
```

```bash
rosemary (SPARC-system) # ./setup_install_server -t /x86S0 /export/home/s9cdx86
```

```bash
rosemary (SPARC-system) # cd /
```

```bash
rosemary (SPARC-system) # unmount /x86S0
```

```bash
rosemary (SPARC-system) # unmount /x86S2
```

```bash
tadpole (remote-x86-system) # unshare /cdrom/cdrom0/s0
```

```bash
tadpole (remote-x86-system) # unshare /cdrom/cdrom0/s2
```

```bash
rosemary (SPARC-system) # cd /cdrom/cdrom0/Solaris_9/Tools
```

```bash
rosemary (SPARC-system) # ./add_to_install_server /export/home/s9cdx86
```

```bash
rosemary (SPARC-system) # cd /cdrom/cdrom0/Tools
```

```bash
rosemary (SPARC-system) # ./add_to_install_server /export/home/s9cdx86
```

On the remote x86 system:

```bash
tadpole (remote-x86-system) # share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
```

```bash
tadpole (remote-x86-system) # share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
```

```bash
tadpole (remote-x86-system) # /etc/init.d/nfs.server start
```

On the SPARC system:

```bash
rosemary (SPARC-system) # mount remote_x86_system_name:/cdrom/sol_9_x86 /x86S0
```

```bash
rosemary (SPARC-system) # mount remote_x86_system_name:/cdrom/sol_9_x86 /x86S2
```

```bash
rosemary (SPARC-system) # cd /x86S2
```

```bash
rosemary (SPARC-system) # ./modify_install_server -p /export/home/s9cdx86 /x86S0
```

Chapter 12 • Preparing to Install From the Network With CD Media (Tasks) 111
EXAMPLE 12-3 Creating an x86 Install Server on a SPARC System With x86 CD Media
(Continued)

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

To Create a SPARC Install Server on an x86 System
With SPARC CD Media

Use this procedure to create a SPARC install server on an x86 system with SPARC CD
media. You need the following:

- An x86 system
- A SPARC system with a CD-ROM drive
- A set of CDs for the remote SPARC system
  - Solaris 9 Software 1 of 2 SPARC Platform Edition CD
  - Solaris 9 Software 2 of 2 SPARC Platform Edition CD
  - Solaris 9 SPARC Platform Edition Languages CD
  - Solaris 9 SPARC Platform Edition Installation CD

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

In this procedure, x86-system is the x86 system that is to be the install server and
remote-SPARC-system is the remote SPARC system to be used with the SPARC CD
media.

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

2. On the remote SPARC system, insert the Solaris 9 Software 1 of 2 SPARC Platform
Edition CD into the system’s drive.

3. On the remote SPARC system, add the following entries to the /etc/dfs/dfstab
file.

```bash
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
```
4. On the remote SPARC system, start the NFS daemon.
   
   ```
   remote-SPARC-system: /etc/init.d/nfs.server start
   ```

5. On the remote SPARC system, verify that the CD is available to other systems by using the `share` command.
   
   ```
   remote-SPARC-system: share
   - /cdrom/cdrom0/s0 ro,anon=0 " "
   - /cdrom/cdrom0/s1 ro,anon=0 " "
   ```

6. On the x86 system that is to be the SPARC install server, become superuser.

7. On the x86 system, access the SPARC CD by creating two directories for the appropriate mount points, one for the miniroot and one for the product.
   
   ```
   x86-system: mkdir directory_name_s0
   x86-system: mkdir directory_name_s1
   ```

   **directory_name_s0**  Is the name of the directory to contain the product from slice 0

   **directory_name_s1**  Is the name of the directory to contain the miniroot from slice 1

8. Verify that the CD is properly exported on the remote x86 system.
   
   ```
   x86-system: showmount -e remote-SPARC-system
   export list for remote-SPARC-system:
   /cdrom/sol_9_sparc/s0 (everyone)
   /cdrom/sol_9_sparc/s1 (everyone)
   ```

9. On the x86 system, mount the remote SPARC CD image.
   
   ```
   x86-system: mount remote_SPARC_system_name: /cdrom/cdrom0/s0 directory_name_s0
   x86-system: mount remote_SPARC_system_name: /cdrom/cdrom0/s1 directory_name_s1
   ```

10. On the x86 system, change to the Tools directory on the mounted disc:
    
    ```
    x86-system: cd /directory_name_s0/Solaris_9/Tools
    ```

11. On the x86 system, copy the disc in the drive to the install server's hard disk in the directory you've created by using the `setup_install_server` command:
    
    ```
    x86-system: ./setup_install_server -t directory_name_s1 install_dir_path
    ```

    **-t**  Specifies the path to a boot image if you want to use a boot image other than the one in the Tools directory on the Solaris 9 Software 2 of 2 CD.

    **directory_name_s1**  Is the name of the directory that contains the miniroot from slice 1.
install_dir_path

Specifies the directory where the disc image is to be copied. The directory must be empty.

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

12. On the x86 system, change to the top directory.
   ```bash
   x86-system# cd /
   ```

13. On the x86 system, unmount both directories.
   ```bash
   x86-system# unmount directory_name_s0
   x86-system# unmount directory_name_s1
   ```

14. On the SPARC system, unshare both CD-ROM slices.
   ```bash
   remote-SPARC-system# unshare /cdrom/cdrom0/s0
   remote-SPARC-system# unshare /cdrom/cdrom0/s1
   ```

15. On the SPARC system, eject the Solaris 9 Software 1 of 2 SPARC Platform Edition CD.

16. Insert the Solaris 9 Software 2 of 2 SPARC Platform Edition CD into the x86 system’s CD-ROM drive.

17. On the x86 system, change to the Tools directory on the mounted CD:
   ```bash
   x86-system# cd /cdrom/cdrom0/Solaris_9/Tools
   ```

18. On the x86 system, copy the CD to the install server’s hard disk:
   ```bash
   x86-system# ./add_to_install_server install_dir_path
   ```

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

19. Eject the Solaris 9 Software 2 of 2 SPARC Platform Edition CD.

20. On the x86 system, insert the Solaris 9 SPARC Platform Edition Languages CD into the x86 system’s CD-ROM drive and mount the CD.

21. On the x86 system, change to the Tools directory on the mounted CD:
   ```bash
   x86-system# cd /cdrom/cdrom0/Tools
   ```

22. On the x86 system, copy the CD to the install server’s hard disk:
23. Decide if you want to enable users to use the Solaris Web Start installation method to boot a system and install the Solaris 9 software from a network.

- If no, eject the Solaris 9 SPARC Platform Edition Languages CD and proceed to Step 31.
- If yes, eject the Solaris 9 SPARC Platform Edition Languages CD and continue.

24. On the remote SPARC system, insert the Solaris 9 SPARC Platform Edition Installation CD into the SPARC system’s CD-ROM drive.

25. On the remote SPARC system, add the following entry to the /etc/dfs/dfstab file.

```bash
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
```

26. On the remote SPARC system, start the NFS daemon.

```bash
remote-SPARC-system# /etc/init.d/nfs.server start
```

27. On the remote SPARC system, verify that the CD is available to other systems:

```bash
remote-SPARC-system# share
- /cdrom/multi_sol_9_x86/s0 ro,anon=0 " "
- /cdrom/multi_sol_9_x86/s1 ro,anon=0 " "
```

In the previous sample output, multi_sol_9_x86 refers to the Solaris 9 (x86 Platform Edition) operating environment. This text string varies for each version of the Solaris operating environment.

28. On the x86 system, mount the slices on the CD.

```bash
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s0 directory_name_s0
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s1 directory_name_s1
```

29. On the x86 system, change to the directory that contains `modify_install_server` on the mounted CD:

```bash
x86-system# cd directory_name_s0
```

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

```bash
x86-system# ./modify_install_server -p install_dir_path directory_name_s1
```

```
-p
```

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

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

**directory_name_s1**

Name of the directory to contain the miniroot from slice 1.

31. Decide if you want to patch the files that are located in the miniroot (Solaris 9/Tools/Boot) on the net install image that was created by setup_install_server.

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

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

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

- If the install server is on the same subnet as the system to be installed or you are using DHCP, you do not need to create a boot server. See “Adding Systems to Be Installed From the Network With a CD Image” on page 119.
- If the install server is not on the same subnet as the system to be installed and you are not using DHCP, you must create a boot server. For detailed instructions on how to create a boot server, refer to “To Create a Boot Server on a Subnet With a CD Image” on page 118.

**EXAMPLE 12-4 Creating a SPARC Install Server on an x86 System With SPARC CD Media**

The following example illustrates how to create a SPARC install server on an x86 system that is named `richards`. The following SPARC CDs are copied from a remote SPARC system that is named `simpson` to the x86 install server’s `/export/home/s9cdsparc` directory.

- Solaris 9 Software 1 of 2 SPARC Platform Edition CD
- Solaris 9 Software 2 of 2 SPARC Platform Edition CD
- Solaris 9 SPARC Platform Edition Languages CD
- Solaris 9 SPARC Platform Edition Installation CD

On the remote SPARC system:

```
simpson (remote-SPARC-system) # share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
simpson (remote-SPARC-system) # share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
simpson (remote-SPARC-system) # /etc/init.d/nfs.server start
```

On the x86 system:

```
richards (x86-system) # mkdir /sparcS0
richards (x86-system) # mkdir /sparcS1
richards (x86-system) # mount simpson:/cdrom/cdrom0/s0 /sparcS0
```
Creating a SPARC Install Server on an x86 System With SPARC CD Media

(Continued)

richards (x86-system)# mount simpson:/cdrom/cdrom0/s1 /sparcS1
richards (x86-system) cd /sparcS0/Solaris_9/Tools
richards (x86-system) ./setup_install_server -t /sparcS0 /export/home/s9cdsparc
richards (x86-system) cd /
richards (x86-system) unmount /sparcS0
richards (x86-system) unmount /sparcS1

On the remote SPARC system:

simpson (remote-SPARC-system) unshare /cdrom/cdrom0/s0
simpson (remote-SPARC-system) unshare /cdrom/cdrom0/s1

On the x86 system:

richards (x86-system)# cd /cdrom/cdrom0/Solaris_9/Tools
richards (x86-system)# ./add_to_install_server /export/home/s9cdsparc
richards (x86-system)# cd /cdrom/cdrom0/Tools
richards (x86-system)# ./add_to_install_server /export/home/s9cdsparc

On the remote SPARC system:

simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
simpson (remote-SPARC-system)# /etc/init.d/nfs.server start

On the x86 system:

richards (x86-system)# mount remote_SPARC_system_name:/cdrom/cdrom0 /sparcS0
richards (x86-system)# mount remote_SPARC_system_name:/cdrom/cdrom0 /sparcS1
richards (x86-system)# cd /sparcS0
richards (x86-system)# ./modify_install_server -p /export/home/s9cdsparc /sparcS1

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

Creating a Boot Server on a Subnet With a CD Image

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

- If you are using DHCP to set installation parameters or your install server and client are on the same subnet, you do not need a boot server. Proceed to “Adding Systems to Be Installed From the Network With a CD Image” on page 119.
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.
   
   ```bash
   # 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.
   
   ```bash
   # 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.
   
   ```bash
   # cd /mnt/Solaris_9/Tools
   ```

5. Copy the boot software to the boot server.
   
   ```bash
   # ./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 -k` command.

6. Change directories to root (`/`).
   
   ```
   # cd /
   ```

7. Unmount the installation image.
   
   ```
   # umount /mnt
   ```

You are now ready to set up systems to be installed from the network. See “Adding Systems to Be Installed From the Network With a CD Image” on page 119

**EXAMPLE 12-5 Creating a Boot Server on a Subnet With CD Media**

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

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

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

---

**Adding Systems to Be Installed From the Network With a CD Image**

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

- An install server
- A boot server if it is required
- The `sysidcfg` file if you use a `sysidcfg` file to preconfigure system information
- A name server if you use a name service to preconfigure system information
- The profile in the JumpStart directory on the profile server if you are using the custom JumpStart installation method
Use the following add_install_client procedure for setting up install servers and clients. Also, see the example procedures for the following:

- If you are using DHCP to set installation parameters, see Example 12–6.
- If your install server and client are on the same subnet, see Example 12–7.
- If your install server and your client are not on the same subnet and you are not using DHCP, see Example 12–8.

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
Specifies that the client is to use DHCP to obtain the network install parameters.

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

Specifies the name and path to the install server.
- s install_server:install_dir_path
  - install_server is the host name of the install server
  - install_dir_path is the absolute path to the Solaris 9 CD image

Specifies a JumpStart directory for custom JumpStart installations. jumpstart_server is the host name of the server on which the JumpStart directory is located. jumpstart_dir_path is the absolute path to the JumpStart directory.
- c jumpstart_server:jumpstart_dir_path

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

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

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

platform_group
Is the platform group of the system to be installed. A detailed list of platform groups appears in Chapter 39.

**EXAMPLE 12-6** SPARC: Adding a SPARC Install Client on a SPARC Install Server When Using DHCP (CDs)

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

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

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

Chapter 12 • Preparing to Install From the Network With CD Media (Tasks) 121
EXAMPLE 12-7 Adding an Install Client That Is on the Same Subnet as Its Server (CDs)

The following example illustrates how to add an install client that is on the same subnet as the install server. The install client is named basil, which is an Ultra™ 5 system. The file system /export/home/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 12-8 Adding an Install Client to a Boot Server (CDs)

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

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

EXAMPLE 12-9 x86: Adding an x86 Install Client on an x86 Install Server When Using DHCP (CDs)

The following example illustrates how to add an x86 install client to an install server when you are using DHCP to set installation parameters on the network. The -d option is used to specify that clients are to use the DHCP protocol for configuration. If you plan to use PXE network boot, you must use the DHCP protocol. The DHCP class name SUNW.i86pc indicates that this command applies to all Solaris x86 network boot clients, not just a single client. The -s option is used to specify that the clients are to be installed from the install server that is named rosemary. This server contains a Solaris 9 Software 1 of 2 x86 Platform Edition CD image in /export/home/s9cdx86.

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

install_server# cd /export/boot/Solaris_9/Tools
install_server# ./add_install_client -d -s rosemary:/export/home/s9cdx86 SUNW.
 i86pc i86pc
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**

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Platform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>showmount -e</td>
<td>All</td>
<td>A command that lists all the shared file systems that are located on a remote host. The <code>showmount(1M)</code> man page contains more information.</td>
</tr>
<tr>
<td>uname -i</td>
<td>All</td>
<td>A command for determining a system’s platform name, for example, SUNW, SPARCstation-5, or i86pc. You might need the system’s platform name when you install the Solaris software. The <code>uname(1)</code> man page contains more information.</td>
</tr>
<tr>
<td>patchadd -C net_install_image</td>
<td>All</td>
<td>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. <strong>Caution</strong> – Don’t use the <code>patchadd -C</code> command unless you have read the Patch Readme instructions or have contacted your local Sun support office.</td>
</tr>
<tr>
<td>reset</td>
<td>SPARC</td>
<td>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>&gt;</code> PROM prompt.</td>
</tr>
<tr>
<td>banner</td>
<td>SPARC</td>
<td>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>&gt;</code> PROM prompt.</td>
</tr>
</tbody>
</table>
CHAPTER 14

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 125
- “SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program” on page 126
- “x86: Performing an Installation or Upgrade With the Solaris Web Start Program” on page 131
- “Solaris Web Start Post-Installation and Upgrade Tasks” on page 138

Solaris Web Start Program GUI or CLI

You can run the Solaris Web Start program with a GUI or with a CLI. For x86 systems, the Device Configuration Assistant is included in the Solaris Web Start program.

- GUI – Requires a local or remote DVD-ROM or CD-ROM drive or network connection, video adapter, keyboard, monitor, and enough memory.
- CLI – Requires a local or remote DVD-ROM or CD-ROM drive or network connection, keyboard, and monitor. You can run the Solaris Web Start installation CLI with the \texttt{tip} command. For more information, see the man page, \texttt{tip(1)}. 125
If the Solaris Web Start program detects a video adapter for the system, it automatically displays the GUI. If the Solaris Web Start program does not detect a video adapter, it automatically displays the CLI. The content and sequence of instructions in both the GUI and the CLI are generally the same.

SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program

You can either install or upgrade the Solaris operating environment by using the Solaris Web Start program, which is on the Solaris 9 SPARC Platform Edition DVD or the Solaris 9 SPARC Platform Edition 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 27 for special requirements.

For the installation, ensure that you have the following:
- For a DVD installation, the Solaris 9 SPARC Platform Edition DVD
- For a CD installation:
  - Solaris 9 SPARC Platform Edition Installation CD.
  - Solaris 9 Software 1 of 2 SPARC Platform Edition CD.
  - Solaris 9 Software 2 of 2 SPARC Platform Edition CD – The installation program prompts you for this CD if necessary.
  - Solaris 9 SPARC Platform Edition Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.
### SPARC: Task Map: Performing a Solaris Web Start Installation

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify system requirements.</td>
<td>Verify that your system meets the requirements to install or upgrade with the Solaris Web Start program.</td>
<td>“System Requirements” on page 27</td>
</tr>
<tr>
<td>Gather the necessary information.</td>
<td>Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>(Optional) Preconfigure system configuration information.</td>
<td>Use the sysidcfg file or the name service to preconfigure installation information (for example, locale) for a system. Preconfiguring system information prevents the installation program from prompting you during the installation.</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>(Upgrade only) Prepare to upgrade the system.</td>
<td>Back up the system.</td>
<td>System Administration Guide: Basic Administration</td>
</tr>
<tr>
<td>(Optional) Set up the system to install from the network.</td>
<td>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.</td>
<td>Chapter 12</td>
</tr>
<tr>
<td>Install or upgrade.</td>
<td>Boot the system and follow the prompts to install or upgrade the Solaris software.</td>
<td>“SPARC: To Perform an Installation or Upgrade With the Solaris Web Start Program” on page 127</td>
</tr>
<tr>
<td>(Upgrade only) Perform post-upgrade tasks.</td>
<td>Correct any local modifications that were lost during the upgrade.</td>
<td>“To Correct Local Modifications After Upgrading” on page 139</td>
</tr>
</tbody>
</table>

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

1. Decide if you want to install the software by using the DVD-ROM or CD-ROM drive or by using a net image.
   - If you’re using a DVD-ROM or CD-ROM drive, insert the Solaris 9 SPARC Platform Edition DVD or Solaris 9 SPARC Platform Edition Installation CD.
   - If you’re using a net installation image, change directories to where the installation media is located. You might need to contact your network administrator for the location. For more information on installing from the network, see Chapter 11 or Chapter 12.

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:

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

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

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

The Solaris Web Start installation begins.

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

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

   If you did not preconfigure the system configuration information, use the “Worksheet for Installation” on page 44 or the “Worksheet for Upgrading” on page 50 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.
You can click on any link in the Kiosk menu.

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

The Installer Questions screen appears.

4. **Decide if you want to reboot the system automatically and if you want to automatically eject the disc.** Click NEXT.

The Specify Media screen appears.

5. **Specify the media you are using to install.**
   - If you are installing a Solaris Flash archive, continue.
   - If you are not installing a Solaris Flash archive, proceed to Step 6.

   a. **Type the information that you are prompted to enter.**
If you selected to install an archive from a DVD, CD, or from an NFS server, the Select Flash Archives panel is displayed.

b. For archives that are stored on a disc or an NFS server, on the Select Flash Archives panel, select one or more Solaris Flash archives to install.

c. On the Flash Archives Summary panel, confirm the selected archives and click Next.

d. On the Additional Flash Archives panel, you can install an additional Solaris Flash archive by specifying the media where the other archive is located. If you do not want to install additional archives, select None and click Next to continue the installation. Proceed to Step 7.

6. Decide if you want to perform an initial installation or an upgrade.

The Solaris Web Start program determines if the system can be upgraded. You must have an existing Solaris root (/) file system. If you are upgrading by using the Solaris 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. If you are NOT installing a Solaris Flash archive, you can also install additional software.

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

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

- /var/sadm/system/logs
- /var/sadm/install/logs
8. If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to “To Correct Local Modifications After Upgrading” on page 139.

x86: Performing an Installation or Upgrade With the Solaris Web Start Program

You can install the Solaris operating environment on your x86 system by using the Solaris Web Start program on the Solaris 9 DVD or 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 27 for special requirements.

Ensure that you have the following media:

- If you are installing from a DVD, use the Solaris 9 *x86 Platform Edition* DVD
- If you are installing from CD media, use the following:
  - Solaris 9 *x86 Platform Edition* Installation CD
  - Solaris 9 Software 1 of 2 *x86 Platform Edition* CD
  - Solaris 9 Software 2 of 2 *x86 Platform Edition* CD – The installation program prompts you for this CD if necessary.
  - Solaris 9 *x86 Platform Edition* Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

**x86: Task Map: Performing a Solaris Web Start Installation**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify system requirements.</td>
<td>Verify that your system meets the requirements to install or upgrade with the Solaris Web Start program.</td>
<td>“System Requirements” on page 27</td>
</tr>
</tbody>
</table>
### TABLE 14–2 x86: Performing a Solaris Web Start Installation Task Map (Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gather the necessary information.</td>
<td>Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>(Upgrade only) Prepare to upgrade the system.</td>
<td>Back up the system.</td>
<td>System Administration Guide: Basic Administration</td>
</tr>
<tr>
<td>(Optional) Set up the system to install from the network.</td>
<td>To install a system from a remote Solaris 9 x86 Platform Edition DVD or Solaris 9 Software x86 Platform Edition CD net installation image, you must set up the system to boot and install from an install server or boot server.</td>
<td>Chapter 12</td>
</tr>
<tr>
<td>Install or upgrade.</td>
<td>Boot the system and follow the prompts to install or upgrade the Solaris software.</td>
<td>“x86: To Perform an Installation or Upgrade With the Solaris Web Start Program” on page 132</td>
</tr>
<tr>
<td>(Upgrade only) Perform post-upgrade task.</td>
<td>Correct any local modifications that were lost during the upgrade.</td>
<td>“To Correct Local Modifications After Upgrading” on page 139</td>
</tr>
</tbody>
</table>

#### x86: To Perform an Installation or Upgrade With the Solaris Web Start Program

1. Decide if you want to install the software by using the DVD-ROM or CD-ROM drive or by using a net installation image.
   - If you’re using a DVD-ROM or CD-ROM drive, insert the Solaris 9 x86 Platform Edition DVD or the Solaris 9 x86 Platform Edition Installation CD.
   - If you’re using a net installation image, change directories to where the installation media is located. You might need to check with your network administrator for the location. For more information on installing from the network, see Chapter 11 or Chapter 12.

2. Decide how to boot the system.
   - If you boot from the Solaris 9 DVD or the Solaris 9 Installation CD, insert the disc. Your system’s BIOS must support booting from a DVD or CD.
   - If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the system’s BIOS setup tool or the network adapter’s configuration setup tool.
   - If you boot from a diskette, insert a Solaris 9 Device Configuration Assistant x86 Platform Edition diskette into the system’s diskette drive.
Note – You can copy the Device Configuration Assistant software to a diskette by the following methods:

- Copy the software to a diskette from the Solaris 9 x86 Platform Edition DVD or Solaris 9 Software 2 of 2 x86 Platform Edition CD by using the procedure in Appendix E.
- Download and copy the software to a diskette from the Solaris Developer Connection at soldc.sun.com/support/drivers/dca_diskettes.

3. Boot the system by shutting it down and then turning it off and on.
   A memory test and hardware detection are executed. The screen refreshes.

4. When the Solaris Device Configuration Assistant screen is displayed, press F2_Continue.
   The Bus Enumeration screen appears with the message:
   Determining bus types and gathering hardware configuration data ...
   The Scanning Devices screen appears. System devices are scanned. When scanning is complete, the Identified Devices screen appears.

5. Decide if you need to make any changes.
   - Press F2_Continue to make no changes.
   - Select changes and press F4.
   The Loading screen is displayed with messages about drivers that are loaded to boot your system. After a few seconds, the Boot Solaris screen appears.

6. On the Boot Solaris screen, select DVD, CD, Net, or Disk, and press F2_Continue.
   The Solaris Web Start program checks the default boot disk for the requirements to install or upgrade the system.
Note – If you are installing by using the Solaris 9 Installation CD, several requirements must be met. If you cannot meet these requirements, use the Solaris Web Start program from the DVD or a net installation image, or the Solaris suninstall program or custom JumpStart to install. These requirements are the following:

- The BIOS and SCSI driver for the default boot disk must support logical block addressing (LBA).
- If the default boot disk does not have a Solaris fdisk partition, you are prompted to create a partition. You are asked if you want to run fdisk and create a Solaris fdisk partition. If you answer Yes, the fdisk user interface opens so you can manually add a Solaris fdisk partition on the disk. If you answer No, the installer exits.
- If the default boot disk does not have an x86 boot partition, the Solaris Web Start program creates a partition for you from a portion of the Solaris fdisk partition. Any data on the Solaris fdisk partition is destroyed. You cannot create this partition manually. If you answer Yes to formatting the disk, the partition is created from the Solaris fdisk partition. If you answer No, you must choose another method of installing.
- If you want to preserve an existing Service partition on your system, select Default when the Solaris Web Start program asks you if you want to customize the fdisk partitions on the default boot disk.
- Later in the installation, you can create, modify, or delete partitions through format panels. But, after the Solaris fdisk partition and x86 boot partition are created, the disk is frozen and you cannot edit the disk. If you have multiple disks, you can edit those disks at the fdisk panels.

The Solaris Web Start program detects each requirement on the default boot disk and prompts you for configuration information that was not found.

7. If you are prompted, answer the system configuration questions.
   - If you preconfigured all of the system configuration information, proceed to Step 8.
   - If you did not preconfigure the system configuration information, use the “Worksheet for Installation” on page 44 or the “Worksheet for Upgrading” on page 50 to help you answer the system configuration questions.

After a few seconds, the Solaris Installation Program screen appears.

   The kdmconfig – Introduction screen appears.

9. Decide if you have enough memory to run the GUI.
   - Press F2_Continue if you have enough memory to display the GUI.
   - Press F4_Bypass if you don’t have enough memory, and the CLI is displayed.

The kdmconfig - View and Edit Window System Configuration screen appears.
If the kdmconfig utility cannot detect the video driver, the kdmconfig utility selects the 640x480 VGA driver. The Solaris Web Start GUI cannot be displayed with the 640x480 VGA driver. As a result, the Solaris Web Start CLI is displayed. To use the Solaris Web Start GUI, use the kdmconfig utility to select the correct video driver for your system.

10. (Optional) Examine the configuration information on the kdmconfig – View and Edit Window System Configuration screen and make any changes you need.

11. When you are finished, select No changes needed – Test/Save and Exit and press F2_Continue.

The kdmconfig Window System Configuration Test screen appears.

12. Press F2_Continue.

The screen refreshes and the kdmconfig Window System Configuration Test palette and pattern screen appears.

13. Move the pointer and examine the colors that are shown on the palette to ensure that they are displayed accurately.

- If the colors are not displayed accurately, click No. If possible, press any key on the keyboard, or wait until kdmconfig exits the kdmconfig Window System Configuration Test screen automatically. Repeat Step 10 through Step 13 until the colors are displayed accurately and you can move the pointer as expected.

- If the colors are displayed accurately, click Yes.

The Solaris Web Start Installation Kiosk and Welcome to Solaris dialog box appear. If your system has insufficient memory, the Kiosk does not display.
FIGURE 14-2 Solaris Web Start Kiosk Menu

You can click on any link in the Kiosk menu.

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

14. If you are prompted, answer any remaining configuration questions. The Installer Questions screen appears.

15. Decide if you want to reboot the system automatically and if you want to automatically eject the disc. Click NEXT.
Note – If you are using PXE to boot from a network, select manual reboot. You must ensure that the system does not boot from the network when it reboots. To disable network boot, during the reboot use the system’s BIOS setup tool or network adapter’s configuration setup tool.

The Specify Media screen appears.

16. Specify the media you are using to install.

- If you are installing a Solaris Flash archive, continue.
- If you are not installing a Solaris Flash archive, proceed to Step 17.

a. Type the information that you are prompted to enter.

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

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

b. For archives that are stored on a disc or an NFS server, on the Select Flash Archives panel, select one or more Solaris Flash archives to install.

c. On the Flash Archives Summary panel, confirm the selected archives and click Next.

d. On the Additional Flash Archives panel, you can install an additional Solaris Flash archive by specifying the media where the other archive is located. If you do not want to install additional archives, select None and click Next to continue the installation. Complete the installation.
When the Solaris Web Start installation program is finished installing the Solaris software, the system reboots automatically or prompts you to reboot manually. After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

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

17. Decide if you want to perform an initial installation or an upgrade.

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

*Note* – If you restored the Service partition on your system before you started the installation, you might not be able to upgrade to the Solaris operating environment.

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

When the Solaris Web Start installation program is finished installing the Solaris software, the system reboots automatically or prompts you to reboot manually. After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

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

19. If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to “To Correct Local Modifications After Upgrading” on page 139.

---

**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 141
- “SPARC: Performing an Installation or Upgrade With the Solaris `suninstall` Program” on page 142
- “x86: Performing an Installation or Upgrade With the Solaris `suninstall` Program” on page 145
- “Solaris `suninstall` Program Post-Upgrade Task” on page 151

Solaris `suninstall` Program

You can run the Solaris `suninstall` program with a CLI. For x86 systems, the Device Configuration Assistant is included in the Solaris `suninstall` program. The Solaris `suninstall` program requires a local or remote CD-ROM drive or network connection, keyboard, and monitor. You can run the Solaris `suninstall` program with the `tip` command. For more information, see the man page, `tip(1)`.

Note – To navigate through the text, use Control-U to move up a page and Control-D to move down a page.
SPARC: Performing an Installation or Upgrade With the Solaris suninstall Program

You can either install or upgrade the Solaris operating environment on your SPARC system by using the Solaris suninstall program on the Solaris 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 SPARC Platform Edition CD.
- Solaris 9 Software 2 of 2 SPARC Platform Edition CD – The installation program prompts you for this CD if necessary.
- Solaris 9 SPARC Platform Edition Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

SPARC: Task Map: Performing a Solaris suninstall Program Installation

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gather the necessary information.</td>
<td>Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>(Upgrade only) Prepare to upgrade the system.</td>
<td>Back up the system.</td>
<td>System Administration Guide: Basic Administration</td>
</tr>
<tr>
<td>(Optional) Preconfigure system configuration information.</td>
<td>Use the sysidcfg file or the name service to preconfigure installation information for a system, for example locale. Preconfiguring system information prevents the installation program from prompting you to supply the information during the installation.</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>(Optional) Set up the system to install from the network.</td>
<td>To install a system from a remote Solaris 9 Software SPARC Platform Edition CD image, you need to set up the system to boot and install from an install server or boot server.</td>
<td>Chapter 12</td>
</tr>
</tbody>
</table>
TABLE 15-1 SPARC: Task Map: Performing a Solaris suninstall Program Installation (Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install or upgrade.</td>
<td>Boot the system and follow the prompts to install or upgrade the Solaris software.</td>
<td>“SPARC: To Perform an Installation or Upgrade With the Solaris suninstall Program” on page 143</td>
</tr>
<tr>
<td>(Upgrade only) Perform post-upgrade task.</td>
<td>Correct any local modifications that were lost during the upgrade.</td>
<td>“Solaris suninstall Program Post-Upgrade Task” on page 151</td>
</tr>
</tbody>
</table>

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

1. Decide if you want to install the software by using a CD-ROM drive or by using a net image.
   - If you’re using a CD-ROM drive, insert the Solaris 9 Software 1 of 2 SPARC Platform Edition CD
   - If you’re using a net installation image, change directories to where the installation media is located, as in the following example. You might need to check with your network administrator for the location. The following command is an example.
     ```bash
     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:
       ```bash
       ok boot cdrom
       ```
     - To boot from an install server on a network, type:
       ```bash
       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 44 or the “Worksheet for Upgrading” on page 50 to help you answer the system configuration questions.

4. Follow the instructions on the screen to install the Solaris software on the system.
If you are not installing a Solaris Flash archive, complete the installation. If you are
upgrading or need to install additional software, proceed to Step 5.

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

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

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

If you are installing a Solaris Flash archive, continue.

a. On the Flash Archive Retrieval Method screen, select the location of the Solaris
Flash archive.

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

b. Type the information that you are prompted to enter.

<table>
<thead>
<tr>
<th>Media Selected</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Specify the URL and proxy information that is needed to access the Solaris Flash archive.</td>
</tr>
<tr>
<td>FTP</td>
<td>Specify the FTP server and the path to the Solaris Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.</td>
</tr>
<tr>
<td>Network File System</td>
<td>Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.</td>
</tr>
<tr>
<td>Local file</td>
<td>Specify the path to the local file system where the Solaris Flash archive is located.</td>
</tr>
<tr>
<td>Local tape</td>
<td>Specify the local tape device and the position on the tape where the Solaris Flash archive is located.</td>
</tr>
<tr>
<td>Local device</td>
<td>Specify the local device, the path to the Solaris Flash archive, and the type of file system on which the Solaris Flash archive is located.</td>
</tr>
</tbody>
</table>

c. On the Flash Archive Selection screen, you can install an additional Solaris
Flash archive by selecting New. If you do not want to install additional archives,
press Continue to complete the installation.

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

After the installation is finished, installation logs are saved in a file. You can find the
installation logs in the following directories:
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 151.

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

x86: Performing an Installation or Upgrade With the Solaris suninstall Program

You can either install or upgrade the Solaris operating environment on your x86 system by using the Solaris suninstall program on the Solaris 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 x86 Platform Edition CD.
- Solaris 9 Software 2 of 2 x86 Platform Edition CD – The installation program prompts you for this CD if necessary.
- Solaris 9 x86 Platform Edition Languages CD – The installation program prompts you for this CD if necessary to support languages for specific geographic regions.

x86: Performing a suninstall Installation Task Map

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gather the necessary information.</td>
<td>Follow the checklist and complete the worksheet to be sure that you have all of the information you need to install the Solaris software.</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>(Upgrade only) Prepare to upgrade the system.</td>
<td>Back up the system.</td>
<td>System Administration Guide: Basic Administration</td>
</tr>
</tbody>
</table>
TABLE 15–2  x86: Performing an Interactive Installation Task Map   (Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) Preconfigure system configuration information.</td>
<td>You can use the <code>sysidcfg</code> file or the name service to preconfigure installation information (for example, <code>locale</code>) for a system so the installation program does not prompt you to supply the information during the installation.</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>(Optional) Set up the system to install from the network.</td>
<td>To install a system from a remote Solaris 9 Software x86 Platform Edition CD image, you need to set up the system to boot and install from an install server or boot server.</td>
<td>Chapter 12</td>
</tr>
<tr>
<td>Install or upgrade.</td>
<td>Boot the system and follow the prompts to install or upgrade the Solaris software.</td>
<td>“x86: To Perform an Installation or Upgrade With the Solaris suninstall Program” on page 146</td>
</tr>
<tr>
<td>(Upgrade only) Perform post-upgrade task.</td>
<td>Correct any local modifications that were lost during the upgrade.</td>
<td>“Solaris suninstall Program Post-Upgrade Task” on page 151</td>
</tr>
</tbody>
</table>

\[
\text{\textbf{\textgreater} x86: To Perform an Installation or Upgrade With the Solaris suninstall Program}
\]

1. Decide if you want to install the software by using the CD-ROM drive or by using a net image.
   - If you’re using a CD-ROM drive, insert the Solaris 9 Software 1 of 2 x86 Platform Edition CD.
   - If you’re using a net image, change directories to where the installation media is located. You might need to check with your network administrator for the location. The following command is an example.
     ```bash
     $ cd /net/install-svr/export/s9/x86
     ```

2. Decide how to boot the system.
   - If you boot from the Solaris 9 Installation CD, insert the CD. Your system’s BIOS must support booting from a CD.
   - If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the system’s BIOS setup tool or the network adapter’s configuration setup tool.
   - If you boot from a diskette, insert the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette into the system’s diskette drive.
**x86 only** – You can copy the Device Configuration Assistant software to a diskette by the following methods:

- Copy the software to a diskette from the Solaris 9 x86 Platform Edition DVD or Solaris 9 Software 2 of 2 x86 Platform Edition CD by using this procedure: Appendix E.
- Download and copy the software to a diskette from the Solaris Developer Connection at soldc.sun.com/support/drivers/dca_diskettes.

3. **Boot the system by shutting it down and then turning it off and on.**

   A memory test and hardware detection are executed. The screen refreshes.

- If you are using the Solaris 9 Software 1 of 2 x86 Platform Edition CD and Solaris 9 Software 2 of 2 x86 Platform Edition CDs, the following message appears.

  SunOS - x86 Platform Edition Primary Boot Subsystem, vsn 2.0

  Then, information similar to the following test appears.

  SunOS Secondary Boot version 3.00

  Solaris x86 Platform Edition Booting System

  Running Configuration Assistant...

- If you are using PXE network boot to boot from the network, the following message appears.

  Solaris network boot...

  Then, information similar to this appears.

  SunOS Secondary Boot version 3.00

  Solaris x86 Platform Edition Booting System

  Running Configuration Assistant...

- If you are using the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette, the following appears.

  Solaris Boot Sector Version 1

  Then, information similar to the following text appears.

  Solaris for x86 - FCS DCB Version 1.242
loading /solaris/boot.bin

The screen refreshes and information similar to the following text appears.

SunOS Secondary Boot version 3.00

Solaris x86 Platform Edition Booting System

Running Configuration Assistant...

4. **When the Solaris Device Configuration Assistant screen appears, press F2_Continue.**
   
The Bus Enumeration screen appears with the message:

   Determining bus types and gathering hardware configuration data ...

   The Scanning Devices screen appears. System devices are scanned. When scanning is complete, the Identified Devices screen appears.

5. **Decide if you need to make any changes.**
   - Press F2_Continue to make no changes.
   - Select changes and press F4.

   The Loading screen appears and contains messages about drivers that are loaded to boot your system. After a few seconds, the Boot Solaris screen appears.

6. **On the Boot Solaris screen, select CD, Net, or Disk and press F2_Continue.**
   
The Solaris suninstall program checks the default boot disk for the requirements to install or upgrade the system.

---

**Note** – If you want to preserve an existing Service partition on your system, choose the appropriate option from the Create Solaris fdisk Partition screen. For more information on preserving the Service partition, see “x86: Change in Default Boot-Disk Partition Layout” on page 30

If you restored the Service partition on your system before you started the installation, you might not be able to upgrade to the Solaris operating environment.

---

7. **If you are prompted, answer the system configuration questions.**
   - If you preconfigured all of the system configuration information, proceed to Step 8.
If you did not preconfigure the system configuration information, use the “Worksheet for Installation” on page 44 or the “Worksheet for Upgrading” on page 50 to help you answer the system configuration questions.

After a few seconds, the Solaris Installation Program screen appears.

8. **Press F2_Continue.**
   The kdmconfig – Introduction screen appears.

9. **Press F2_Continue.**
   The kdmconfig - View and Edit Window System Configuration screen appears.

10. **(Optional) Examine the configuration information on the kdmconfig - View and Edit Window System Configuration screen and make any changes you need.**

11. **When you are finished, select No changes needed - Test/Save and Exit, and press F2_Continue.**
    The kdmconfig Window System Configuration Test screen appears.

12. **Press F2_Continue.**
    The screen refreshes and the kdmconfig Window System Configuration Test palette and pattern screen appears.

13. **Move the pointer and examine the colors that are shown on the palette to ensure that they are displayed accurately.**
    - If the colors do not display accurately, click No. If possible, press any key on the keyboard, or wait until kdmconfig exits the kdmconfig Window System Configuration Test screen automatically. Repeat Step 10 through Step 13 until the colors are displayed accurately and you can move the pointer as expected.
    - If the colors display accurately, click Yes.
    The Solaris suninstall program begins.

14. **If you are prompted, answer any remaining system configuration questions.**
    - If you preconfigured all of the system configuration information, the Solaris suninstall program does not prompt you to enter any configuration information.
    - If you did not preconfigure the system configuration information, use the “Worksheet for Installation” on page 44 or the “Worksheet for Upgrading” on page 50 to help you answer the system configuration questions.

15. **Decide if you want to reboot the system automatically and if you want to automatically eject the disc.**
Note – If you are using PXE from a network to boot, select manual reboot. You must ensure that the system does not boot from the network when it reboots. To disable network boot, during the reboot use the system’s BIOS setup tool or network adapter’s configuration setup tool.

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

- If you are not installing a Solaris Flash archive, complete the installation. If you are upgrading or need to install additional software, proceed to Step 17.
  When the Solaris suninstall program finishes installing the Solaris software, the system reboots automatically or prompts you to reboot manually.
  After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:
  - /var/sadm/system/logs
  - /var/sadm/install/logs
- If you are installing a Solaris Flash archive, continue.

a. On the Flash Archive Retrieval Method screen, select the location of the Solaris Flash archive.

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

b. Type the information that you are prompted to enter.

<table>
<thead>
<tr>
<th>Media Selected</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Specify the URL and proxy information that is needed to access the Solaris Flash archive.</td>
</tr>
<tr>
<td>FTP</td>
<td>Specify the FTP server and the path to the Solaris Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.</td>
</tr>
<tr>
<td>Network File System</td>
<td>Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.</td>
</tr>
<tr>
<td>Local file</td>
<td>Specify the path to the local file system where the Solaris Flash archive is located.</td>
</tr>
<tr>
<td>Local tape</td>
<td>Specify the local tape device and the position on the tape where the Solaris Flash archive is located.</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Media Selected</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local device</td>
<td>Specify the local device, the path to the Solaris Flash archive, and the type of file system on which the Solaris Flash archive is located.</td>
</tr>
</tbody>
</table>

c. On the Flash Archive Selection screen, you can install an additional Solaris Flash archive by selecting New. If you do not want to install additional archives, press Continue to complete the installation.

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

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

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

17. If you are upgrading the Solaris software, you might need to correct some local modifications that were not preserved. For detailed instructions, refer to “Solaris suninstall Program Post-Upgrade Task” on page 151.

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

---

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
Solaris Flash Installation Feature
(Topics)

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

| Chapter 17 | Provides overview and planning information on creating and installing Solaris Flash archives. |
| Chapter 18 | Provides step-by-step instructions for creating Solaris Flash archives. |
| Chapter 19 | Provides references to instructions for using Solaris Flash archives to install systems. Also provides instructions for administering archives. |
| Chapter 20 | Describes syntax and options for the `flar create` command. Describes keywords for Solaris Flash archive information. |
Solaris Flash (Overview and Planning)

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

- “Solaris Flash Introduction” on page 155
- “Planning Your Solaris Flash Installation” on page 160

Solaris Flash Introduction

The Solaris Flash installation feature enables you to use a single reference installation of the Solaris operating environment on a system, which is called the master system. Then, you can replicate that installation on a number of systems, which are called clone systems. You can replicate clone systems with a Solaris Flash initial installation that overwrites all files on the system or with a Solaris Flash update that only includes the differences from two system images. A differential update changes only the files that are specified and is restricted to systems that contain software consistent with the old master image.

Installing Clone Systems With an Initial Installation

You can install a master system with a Solaris Flash archive for an initial installation by using any installation method: Solaris Web Start, custom JumpStart, or Solaris Live Upgrade. All files are overwritten. The Solaris Flash installation is a five–part process.

1. Install the master system. You select a system and use any of the Solaris installation methods to install the Solaris operating environment and any other software.
2. (Optional) Prepare customization scripts to reconfigure or customize the clone system before or after installation.
3. Create the Solaris Flash archive. The Solaris Flash archive contains a copy of all of the files on the master system, unless you excluded some nonessential files.

4. Install the Solaris Flash archive on clone systems. When you install the Solaris Flash archive on a system, all of the files in the archive are copied to that system. The newly installed system now has the same installation configuration as the original master system, thus the system is called a clone system. Some customization is possible with the use of scripts.

5. (Optional) Save a copy of the master image. If you plan to create a differential archive, the master image must be available and identical to the image installed on the clone systems.

For step-by-step instructions, see “Installing the Master System” on page 170.

Figure 17–1 shows an installation of clone systems with an initial installation. All files are overwritten.
Updating Clone Systems With a Solaris Flash Differential Archive

If you have a clone system and want to update it, you can create a differential archive that contains only the differences between two images, the unchanged master image and an updated master image. When you update a clone system with a differential archive, only the files that are in the differential archive are changed. You use the Solaris Flash custom JumpStart installation method to install an update on a clone system. An update is a five-part process.

1. Prepare the master system with changes. Before changes are made, the master system should be running a duplicate of the original archive.
Note – If the master system is not running a duplicate of the original archive, the differences between the two system images might result in a large differential archive. Consequently, installing the differential archive could be time consuming. Use an initial installation with a full archive in this case.

2. (Optional) Prepare customization scripts to reconfigure or customize the clone system before or after installation.

3. Mount the directory of a copy of the saved-unchanged master image. This second image is to be used to compare the two system images. Access the image by the following methods:
   - Mounted from a Solaris Live Upgrade boot environment
   - Mounted from a clone system over NFS
   - Restored from backup using the ufsrestore command.

4. Create the differential archive with the -A option of the flarcreate command.

5. Install the differential archive on clone systems with custom JumpStart.

Figure 17–2 shows the creation and installation of a differential archive. A master image is updated with some modifications. These modifications could be as simple as the addition, reconfiguration, or deletion of a few files, or as complex as propagating patches. The updated master image is compared to the unchanged master image. The differences between the two images become the differential archive. The archive can be used to update other clone systems that are currently using the unchanged master image. If the clone system has already been modified or is not running the unchanged master image, the update fails. If you have many changes to make on the clone systems, you can do an initial installation at any time.
Install clone systems with modifications and validate master and clone system images.

- Exact duplicate of master
- Duplicate of master but contains small changes
- Duplicate of master but contains extra files
- Duplicate of master but missing some files
- Created from different master or installed separately
- Update fails

**FIGURE 17-2 Solaris Flash Update**

Chapter 17 • Solaris Flash (Overview and Planning)
Planning Your Solaris Flash Installation

Before you create and install a Solaris Flash archive, you must make some decisions about how you want to install the Solaris operating environment on your systems. The first time you install a system, you need to install with a full archive that is an initial installation. After a system has been installed with an archive, the system can be updated with a differential archive. The differential archive installs only the differences between two archives.

Designing an Initial Installation of the Master System

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

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

You must install the master system with the exact configuration that you want on each of the clone systems. The decisions that you make when you design the installation of the master system depend on the following:

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

The software on the system can then be customized. See “Customizing an Archive’s Files and Directories” on page 164. The master system can also be modified and set up for an update of a clone system. See “Planning the Creation of a Solaris Flash Archive” on page 162.

Customizing the Solaris Installation on the Master System

After you install the Solaris operating environment on the master system by using any of the Solaris installation methods, you can add or delete software and modify system configuration information as necessary.
Delete software. You can remove software that you determine is not necessary to install on the clone systems. To see a list of software that is installed on the master system, use the Product Registry. For detailed instructions, refer to System Administration Guide: Basic Administration.

Add software. You can install software that is included in the Solaris release. You can also add software that is not delivered as part of the Solaris operating environment. All of the software that you install on the master system is included in the Solaris Flash archive and is installed on the clone systems.

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

Further customization can be done when creating the archive. For example, you can exclude large data files that you might not want in the archive. For an overview, see “Customizing an Archive’s Files and Directories” on page 164.

To update a clone system, you can create a differential archive, see “Planning the Creation of a Solaris Flash Archive” on page 162.

Creating Archives for SPARC and x86 Systems

If you want to use the Solaris Flash installation method to install the Solaris software on both SPARC and x86 systems, you must create a separate Solaris Flash archive for each platform. Use the Solaris Flash archive that was created from the SPARC master system to install SPARC systems. Use the Solaris Flash archive that was created from the x86 master system to install x86 systems.

SPARC: Supporting Peripheral Devices Not Found on the Master System

Choosing the drivers to install on the master system depends on the type of peripheral devices attached to both the master system and the clone system and depends on the type of software group installed. If the clone systems have different peripheral devices than the master system, you need to install the appropriate drivers on the master system before you create the archive.

Note – When you install Solaris software, some hardware-specific drivers are installed only if their supported hardware is found on the system.

If you install the master system with the Core, End User, Developer, or Entire Software Group, the master system supports only the peripheral devices that are attached to the master system at the time of installation. You can install support for peripheral devices
on the master system even though the master system does not have the devices by installing the master system with the Entire Plus OEM Software Group. All drivers are installed regardless of the hardware present on the system. Or, you can install selected packages that support the peripherals needed.

If you install any software group except the Entire Plus OEM Software Group on your master system, you might not have all the drivers that you need to correctly install your clone systems. For example, if you install the Entire Software Group on a master system that has a GX CG6 frame buffer, only the GX CG6 frame buffer driver is installed. This situation is not a problem if all the clone systems you want to install have the GX CG6 frame buffer or no frame buffer.

If you want to install the archive on clone systems that have other frame buffers installed, such as the Elite 3D, then the clone system does not have frame buffer support. Therefore, you must create an archive that includes all frame buffers by installing the Entire Plus OEM Software Group or installing the frame buffer drivers you want by adding the appropriate frame buffer driver packages to the master system.

You can install support for peripherals on clone systems that are different from the master system in one of the following ways.

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

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

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

### Planning the Creation of a Solaris Flash Archive

You can create an archive from the master system or create a differential archive from two system images. The differential archive installs only the differences between the two images.
Planning the Solaris Flash Archive From the Master System for an Initial Installation

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

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

Planning To Create the Solaris Flash Differential Archive

If you have a clone system and want to update it, you can create a differential archive that contains only the differences between two images, the unchanged master image and an updated master image. One image can be your system that is running the original software that was installed on the clone, or you need to access a saved copy of the unchanged master image. This image is to be updated with the desired changes. Another image is to be accessed and used for comparison. The differences between these two images is the differential archive. You use the Solaris Flash custom JumpStart installation method to install an update on a clone system. After you update a clone system with a differential archive, only the files that are in the differential archive are changed on the clone system. Scripts can be used to customize the archive before or after installation, which is especially helpful for reconfiguration.

An unchanged master image should be saved after the initial installation so it could be accessed by any of the following methods.

- A Solaris Live Upgrade boot environment, mounted on some directory that uses the `lumount` command. For a description of a Solaris Live Upgrade boot environment, see Chapter 30
- A clone system that is mounted over NFS with root permissions.
- A system backup that can be restored with the `ufsdump` command.

For step-by-step instructions, see “To Update a Master Image and Create a Differential Archive” on page 180
Customizing an Archive’s Files and Directories

When you create a Solaris Flash archive, some files and directories that are to be copied from the master system can be excluded. If you have excluded a directory, you can also restore specified files or subdirectories under that directory. For example, you could create an archive that excludes all files and directories in /a/a/a/b/b/c. The content of the b b subdirectory could be included. The only content then would be in the b b subdirectory.

Caution – Use the flar create file-exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.

The following table lists the flar create command options that can exclude files and directories and restore files and subdirectories.

<table>
<thead>
<tr>
<th>How Specified?</th>
<th>Options That Exclude</th>
<th>Options That Include</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the name of the directory or file</td>
<td>-x exclude_dir/file</td>
<td>-y include_dir/file</td>
</tr>
<tr>
<td>Use a file that contains a list</td>
<td>-X list_filename</td>
<td>-Z list_filename</td>
</tr>
</tbody>
</table>

For descriptions of these options, see Table 20–7.

For examples of customizing an archive, see “Examples — Creating an Archive for an Initial Install” on page 176.

Customizing an Archive With Scripts

After the software is installed on the master system, special scripts can be run during creation, installation, postinstallation and first reboot. These scripts enable you to do the following:

- Configure applications on clone systems. You can use a custom JumpStart script for some uncomplicated configurations. For more complicated configurations, special configuration file processing might be necessary on the master system or before or after installation on the clone system.
- Protect local customizations on clone systems. Local preinstallation and postinstallation scripts reside on the clone and protect local customizations from being overwritten by the Solaris Flash software.
Identify nonclonable, host-dependent data that enables you to make the archive host independent. Host independence is enabled by modifying such data or excluding it from the archive. An example of host-dependent data is a log file.

- Validate software integrity in the archive during creation.
- Validate the installation on the clone system.

**Guidelines for Creating a Custom Script**

When creating a script other than the reboot script, following these guidelines to assure the script does not corrupt the operating environment or otherwise disrupt the system. These guidelines enable the use of Solaris Live Upgrade, which creates a new boot environment for installation of the operating environment. The new boot environment can be installed with an archive while the current system is running.

**Note** – These guidelines are not for reboot scripts that are allowed to run daemons or make other types of modification to the root (/) file system.

- Scripts must not affect the currently running system. The currently running operating environment might not be the one running when the Solaris Flash archive is installed.
- Scripts must not start or stop any daemon processes.
- Scripts must not depend on the output of commands such as `ps`, `truss`, or `uname`, which are dependent on the operating environment. These commands report information about the currently running system.
- Scripts must not send any signals or otherwise affect any currently-running processes.
- Scripts can use standard UNIX commands that facilitate shell scripting such as `expr`, `cp`, and `ls`.

For an overview of Solaris Live Upgrade, see Chapter 30

**Solaris Flash Archive Sections**

Solaris Flash archives contain the following sections. Some sections can be used by you to identify and customize the archive and view status information on the installation. For a further description of each section, see Chapter 20.
<table>
<thead>
<tr>
<th>Section Name</th>
<th>Informational Only</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive cookie</td>
<td>X</td>
<td>The first section contains a cookie that identifies the file as a Solaris Flash archive.</td>
</tr>
<tr>
<td>Archive identification</td>
<td></td>
<td>The second section contains keywords with values that provide identification information about the archive. Some identification information is supplied by the archive software. Other specific identification information can be added by you by using options to the <code>flar create</code> command.</td>
</tr>
<tr>
<td>User-defined</td>
<td></td>
<td>This section follows the identification section. You can define and insert these sections to customize the archive. The Solaris Flash archive does not process any sections that you insert. For example, a section could contain a description of the archive or perhaps a script to check the integrity of an application.</td>
</tr>
<tr>
<td>Manifest</td>
<td>X</td>
<td>This section is produced for a Solaris Flash differential archive and used for validating a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. This section is informational only, lists the files in an internal format, and cannot be used for scripting.</td>
</tr>
<tr>
<td>Predeployment, Postdeployment, Reboot</td>
<td>X</td>
<td>This section contains internal information that the flash software uses before and after installing an operating environment image. Any scripts that you have provided are included in this section.</td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td>This section contains messages about the archive creation. The section also records the activities of predeployment and postdeployment scripts. You can view the success of the installation in this section by writing a script to send output to this section.</td>
</tr>
<tr>
<td>Archive files</td>
<td>X</td>
<td>The archive files section contains the files that have been gathered from the master system.</td>
</tr>
</tbody>
</table>

When to Create the Archive for an Initial Installation

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

Create the archive after software is installed on the master system and before software is configured. For example, Solaris Volume Manager stores meta-information outside the file system. If you have an encapsulated and mirrored boot disk, the creation of the archive is not possible. Because archive creation does not access the meta-information outside the file system, the archive must be created before configuration.
Where to Store the Solaris Flash Archive

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

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

Compressing the Archive

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

Planning the Installation of Solaris Flash Archives

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

Deciding How to Install Solaris Flash Archives

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

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

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

For installation instructions, see “SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program” on page 126 or “x86: Performing an Installation or Upgrade With the Solaris Web Start Program” on page 131.

The Solaris `suninstall` program on the Solaris 9 Software 1 of 2 CD enables you to install Solaris Flash archives that are stored on the following:
- NFS server
- HTTP server
- FTP server
- Local file
- Local tape
- Local device, including CD

For installation instructions, see “SPARC: Performing an Installation or Upgrade With the Solaris suninstall Program” on page 142 or “x86: Performing an Installation or Upgrade With the Solaris suninstall Program” on page 145.

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

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

For installation instructions, see “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 257.

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

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

For installation instructions, see “Installing Solaris Flash Archives on a Boot Environment” on page 372.
Creating Solaris Flash Archives (Tasks)

This chapter provides the procedures for creating a Solaris Flash archive. These procedures include installing a master system and then creating a Solaris Flash archive from that master system. You can also create a differential archive if you have previously installed an archive on a clone system. When the differential archive is created, two images are compared: the unchanged master image and an updated image. The differential archive installs only the differences between the two images. Also, procedures to create scripts are provided to reconfigure or otherwise customize the archive.

- “Task Map: Creating Solaris Flash Archives” on page 169
- “Installing the Master System” on page 170
- “Creating Customization Scripts” on page 171
- “Creating a Solaris Flash Archive” on page 175

Task Map: Creating Solaris Flash Archives

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install your chosen software configuration on the master system.</td>
<td>Determine the configuration that meets your needs and use any of the Solaris installation methods to install the master system</td>
<td>“To Install the Master System for an Initial Installation” on page 171</td>
</tr>
</tbody>
</table>
TABLE 18–1 Task Map: Creating a Solaris Flash Archive to Install With an Initial Installation  

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
</table>
| **(Optional) Create customization scripts** | Determine if you need to create scripts to do the following:  
  - Customize or reconfigure the archive  
  - Protect local changes on clone systems | “Creating Customization Scripts” on page 171                                      |
| Create the Solaris Flash archive.         | Use the `flar create` command to create an archive                           | “To Create a Solaris Flash Archive for an Initial Installation” on page 176       |
| **(Optional) Save a copy of the archive**  | Keep a copy of the archive for future comparison to update a clone system with a differential archive | “To Create a Solaris Flash Archive for an Initial Installation” on page 176       |

TABLE 18–2 Task Map: Creating a Solaris Flash Archive to Update a Clone System

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare master image</td>
<td>Make changes to the unchanged master image such as adding or deleting packages or installing patches</td>
<td>“Installing the Master System” on page 170</td>
</tr>
</tbody>
</table>
| **(Optional) Create customization scripts** | Determine if you need to create scripts to do the following:  
  - Customize or reconfigure the archive  
  - Protect local changes on clone systems | “Creating Customization Scripts” on page 171                                      |
| Create the Solaris Flash differential archive | 1. Mount the unchanged master image  
  2. Use the `flar create` command to compare the two images and create the differential archive | “To Update a Master Image and Create a Differential Archive” on page 180          |

Installing the Master System

You install the master system with the software configuration that you want other systems to have. You can install clone systems with an initial installation that overwrites all files on the system or with an update that only includes the differences between two images. For an initial installation, use any of the Solaris installation methods to install the Solaris operating environment on the master system.
If you have previously installed an archive on a clone system, you can update that system with changes by using a differential archive. The changes are made to the original image such as installing patches, or adding and removing packages. The differential archive overwrites only the files specified in the archive. For the procedure for updating the original master image and creating a differential archive, see “To Update a Master Image and Create a Differential Archive” on page 180.

▼ To Install the Master System for an Initial Installation

1. Identify the system configuration that you want to install.

2. With the use of the Solaris installation methods, install the Solaris operating environment on the master system. For a discussion of the different installation methods, refer to Chapter 3.

3. Customize your Solaris installation in any of the following ways:
   - Delete software.
   - Add software.
   - Modify configuration files.
   - Add support for peripheral devices on the clone system.

You can create custom scripts or use the `flar create` command to create the archive.

   - To create custom scripts, see “Creating Customization Scripts” on page 171.
   - To create the archive, see “Creating a Solaris Flash Archive” on page 175.

Creating Customization Scripts

Scripts can customize the archive. Use these scripts for the following purposes:

- A precreation script validates the archive at creation time and prepares the archive for later customization, especially differential archives. This script also can create a user-defined section in the archive.

- A predeployment script validates the archive during installation and prepares the archive for later customization.

- A postdeployment script reconfigures a new system image on a clone system.

- A reboot script processes a final reconfiguration after the system is rebooted.

For guidelines on creating scripts, see “Guidelines for Creating a Custom Script” on page 165.
To Create a Precreation Script

This script runs during archive creation. The script has various uses.

- Validates the contents and the integrity of the software. The script fails the archive creation if the integrity is broken.
- Prepares products for later customization on clone system.
- Registers other installation scripts dynamically during archive creation.
- Adds a message to the flash-creation summary file. The message must be short and record only that scripts were started and finished and the results. You can view the results in the summary section.

1. Create the precreation script. Follow the guidelines that are described in “Guidelines for Creating a Custom Script” on page 165.

2. Store the script in the `/etc/flash/precreation` directory.

**EXAMPLE 18-1 Excerpts From a Precreation Script**

The following examples are excerpts from a precreation script.

- To log the start time in the summary section, use the following example:
  ```bash
  echo "MyApp precreation script started">> $FLASHDIR/summary
  ```
- To check the software integrity, use the `flcheck` command. This command cannot be used at the command line. The syntax for this command is as follows:
  ```bash
  flcheck software component files and directories ...| -
  ```

For example, to validate the files and directories, use the following example:

```bash
flcheck software component files and directories
If Not in selection - refuse creation
```

```bash
echo "Myapp Integrity Damage">>$FlashDIR/summary
```

Or, to keep new files and directories that are unexpected and not fail the archive creation, use the following example:

```bash
flcheck software component files and directories
If Not in selection include by force
flinclude software component
```

- To register deployment scripts and data, use the following example:
  ```bash
  cp predeployment script $FLASHDIR/preinstall/proc.d
  ln -s $FLASHDIR/preinstall/order.ds55.MyApp
  cp postdeployment script $FLASHDIR/postinstall/proc.d
  ln -s $FLASHDIR/postinstall/order.ds55.MyApp
  ```

- To see application specific data in a user-defined section, use the following example:
  ```bash
  cp custom section $FLASHDIR/custom_sections/MyApp
  ```
To log the success of the installation in the summary section, use the following example:

```
echo "product one flash preparation started." >>$FLASH_DIR/summary
```

```
... echo "product one flash preparation finished successfully" >>$FLASH_DIR/summary
```

**EXAMPLE 18-2 Precreation Script**

```
#!/bin/sh

echo "Test precreation script started" >> $FLASH_DIR/summary
cat /opt/TestApp/critical_file_list | flcheck -
if [ $? != 0 ]; then
    echo "Test precreation script failure" >> $FLASH_DIR/summary
    exit 1
fi

echo "Test precreation script started" >> $FLASH_DIR/summary

/opt/TestApplication/license_cloning
    $FLASH_DIR/predeployment/.TestApplicationLicenceTransfer \ 
    $FLASH_DIR/custom_sections/TestApplicationLicenceCounter

echo "Test precreation script finished" >> $FLASH_DIR/summary
exit 0
```

**Using a Precreation Script to Create a User-Defined Archive Section**

A precreation script can create a user-defined section in the archive to provide specific application information. This section is intended for archive maintenance. The script must be put in the `$FLASH_DIR/sections` directory. The Solaris Flash archive does not process a user-defined section. For example, a section could contain a description of the archive or perhaps a script to check the integrity of an application.

A user-defined section requires the following format.

- Must be line oriented
- Must terminate with newline (ASCII 0x0a) characters
- Can have unlimited length of individual lines
- Encode binary data by using base64 or a similar algorithm.

**To Create a Predeployment Script**

This script is run before the installation of the archive. If the script is meant to validate the archive, it is kept in the archive. If the script is meant to preserve local configurations on the clone system, it is kept on the clone system. This script also can analyze and collect local data necessary for later customization. For example, client-specific information can be saved before being overwritten by files about to be extracted. This information can then be used in the final stage after extraction.
1. Create the predeployment script. Follow the guidelines that are described in “Guidelines for Creating a Custom Script” on page 165.

2. Store the script in one of the following directories.
   - For archive validation, store in the /etc/flash/predeployment directory.
   - If you are referencing with a precreation script, store in the $FLASH_DIR/preinstall directory.
   - If you are preserving configurations on a clone system, provide the path to the script that is stored on the clone system with the local_customization keyword in the JumpStart profile.

**EXAMPLE 18–3 Predeployment Script**

```bash
#!/bin/sh
$FLASH_DIR/TestApplication/check_hardware
if [ $? != 0 ]; then
echo Unsupported hardware
exit 1
fi
$FLASH_DIR/TestApplication/check_licence_key
if [ $? != 0 ]; then
echo No license for this host
exit 1
fi
$FLASH_DIR/TestApplication/deploy_license_key
   $FLASH_DIR/TestApplication/.TestApplicationLicenceTransfer
$FLASH_DIR/TestApplication/save_data_files $FLASH_DIR/flash
```

**To Create a Postdeployment Script**

This script is kept in the archive or stored in a local directory on the clone system and runs after installation. The script reconfigures a new system image on a clone system. If the script is stored in the archive, the changes affect all the clone systems. If the script is stored in a local directory on the clone system, the changes affect only the clone system. For example, client-specific information that is saved by a predeployment script can be applied to the clone environment, completing the installation.

Postdeployment scripts can also be used to clean up files after the archive is installed. For example, log files such as those files in /var/adm can be cleaned out.

**Note** – Not all log files need a script for cleanup. Log files in /var/tmp can be excluded when the archive is created.
1. Create the postdeployment script. Follow the guidelines described in “Guidelines for Creating a Custom Script” on page 165.

2. Store the script in one of the following directories.
   - To affect all clone systems, store the script in the /etc/flash/postdeployment directory.
   - To affect only a local clone system, provide the path to the script that is stored on the clone system with the local_customization keyword in the JumpStart profile.

   **EXAMPLE 18-4** Postdeployment Script

   ```sh
   #!/bin/sh
   $FLASH_DIR/TestApplication/clone_reconfiguration
   $FLASH_DIR/TestApplication/restore_data $FLASH_DIR/flash
   ```

   ▼ To Create a Reboot Script

   This script is kept in the archive and runs after the system is rebooted. The script does any final configuration after system reconfiguration.

   After you install the Solaris Flash archive on a clone system, some host-specific files are deleted and are re-created for the clone machine. The installation program uses the sys-unconfig(1M) command and the sysidtool(1M) programs to delete and re-create host-specific network configuration files. The files that are re-created include, for example, /etc/hosts, /etc/defaultrouter, and /etc/defaultdomain. You can use the reboot script for any final reconfiguration.

   1. Create the reboot script.

   2. Store the script in the /etc/flash/reboot directory.

   **EXAMPLE 18-5** Creating a Reboot Script

   ```sh
   #!/bin/sh
   $FLASH_DIR/TestApplication/finalize_license
   ```

---

**Creating a Solaris Flash Archive**

You can create an archive with an initial installation that overwrites all the files on the clone system, or you can create a differential archive that only overwrites the changes that are specified. For an overview of a differential archive, see “Planning To Create the Solaris Flash Differential Archive” on page 163.
To Create a Solaris Flash Archive for an Initial Installation

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

1. **Boot the master system and run it in as inactive a state as possible.**
   When possible, run the system in single-user mode. If that is not possible, shut down any applications that you want to archive and any applications that require extensive operating system resources.
   You can create a Solaris Flash archive while the master system is running in multiuser mode, single-user mode, or while booted from one of the following:
   - Solaris 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 `flar create` command.**
   ```bash
   # flar create -n name options path/filename
   ```
   - `name` The name that you give the archive. The `name` you specify is the value of the `content_name` keyword.
   - `options` For a description of options, see “`flar create`” on page 198.
   - `path` The path to the directory in which you want to save the archive file.
     If you do not specify a path, `flar create` saves the archive file in the current directory.
   - `filename` The name of the archive file.

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

3. **Make a copy of the archive and save it. The copy can be used in the future to update a clone system with a differential archive.**

Examples — Creating an Archive for an Initial Install

File systems can be copied exactly or can be customized by excluding some directories or files. You can achieve the same results by using different options. Use the options that best suit your environment.
The file systems in the following examples have been greatly simplified for clarification. Rather than use file system names such as /var, /usr, or /opt, the master system file structure for these examples is the following:

```
/aaa/bbb/ccc/ddd
/aaa/bbb/fff
/aaa/eee
/ggg
```

**Caution** – Use the `flar create` file exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.

---

**EXAMPLE 18-6 Creating an Exact Duplicate Archive**

In this example, the archive is named `archive1`. It is copied exactly from the master system and then compressed. The archive is an exact duplicate of the master system and stored in `archive1.flar`.

```
# flar create -n archive1 -c archive1.flar
```

To check on the file structure of the archive, type the following.

```
# flar info -l archive1.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb/fff
aaa/eee
aaa/eee
/ggg
```

**EXAMPLE 18-7 Creating an Archive and Excluding and Including Files and Directories**

In this example, the archive is named `archive2`. It is copied from the master system but is not an exact copy. The content under the `/aaa` directory is excluded, but the content in `/aaa/bbb/ccc` remains.

```
# flar create -n archive2 -x /aaa -y /aaa/bbb/ccc archive2.flar
```

To check on the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l aaa
aaa
aaa/bbb/ccc
aaa/bbb/ccc/ddd
```
EXAMPLE 18-7 Creating an Archive and Excluding and Including Files and Directories
(Continued)

aaa/bbb
ggg

EXAMPLE 18-8 Creating an Archive Excluding and Including Files and Directories by Using Lists

In this example, the archive is named archive5. It is copied from the master system but is not an exact copy.

The exclude file contains the following list:
/aaa

The include file contains the following list:
/aaa/bbb/ccc

The content under the /aaa directory is excluded, but the content in /aaa/bbb/ccc remains.

# flar create -n archive5 -X exclude -f include archive5.flar

To check on the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

# flar info -l archive5.flar
aaa
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb
ggg

EXAMPLE 18-9 Creating an Archive Excluding Files and Directories by Using a List and Restoring a Directory

You can combine options -x, -y, -X and -f. In this example, options -X and -y are combined. The archive is named archive5. It is copied from the master system but is not an exact copy.

The exclude file contains the following list:
/aaa

The -y option restores the ccc directory. The following command produces the archive.

# flar create -n archive5 -X exclude -y /ccc archive5.flar

To check on the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.
Creating an Archive Excluding Files and Directories by Using a List and Restoring a Directory (Continued)

# flar info -l archive5.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
999

Creating an Archive Excluding and Including Files and Directories by Using a List With the -z Option

In this example, the archive is named archive3. It is copied from the master system but is not an exact copy. The files and directories to be selected are included in filter1 file. Within the files, the directories are marked with a minus (-) or a plus (+) to indicate which files to exclude and restore. In this example, the directory /aaa is excluded with a minus and the subdirectory /aaa/bbb/ccc is restored with a plus. The filter1 file contains the following list.
- /aaa
+ /aaa/bbb/ccc

The following command produces the archive.

# flar create -n archive3 -z filter1 archive3.flar

To check on the file structure of the archive, type the following command. The excluded directories that include copied files appear, but only the files that were restored contain data.

# flar info -l archive3.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
999

Creating an Archive From an Alternate Boot Environment

In this example, the archive is named archive4. It is copied exactly from the master system and then compressed. The archive is an exact duplicate of the master system and stored in archive4.flar. The -R option is used to create the archive from another directory tree.

# flar create -n archive4 -c -R /x/yy/zz archive4.flar

Creating an Archive and Adding Keywords to Describe the Archive

In this example, the archive is named archive3. This archive is copied exactly from the master system and then compressed. Options add descriptions to the archive identification section, which can help you to identify the archive later. For information on keywords, their values, and formats, see “Solaris Flash Keywords” on page 193.
EXAMPLE 18–12 Creating an Archive and Adding Keywords to Describe the Archive
(Continued)

# flar create -n archive3 -i 20000131221409 -m pumbaa \ 
- e "Solaris 8 Print Server" -a "Mighty Matt" -U "Internal Finance" \ 
- T server archive3.flar

After the archive is created, you can access the identification section that contains the detailed description. An example of an identification section follows.

section_begin=identification
  files_archived_method=cpio
  files_compressed_method=compress
  files_archived_size=259323342
  files_unarchived_size=591238111
  creation_date=20000131221409
  creation_master=pumbaa
  content_name=Finance Print Server
  content_type=server
  content_description=Solaris 8 Print Server
  content_author=Mighty Matt
  content_architectures=sun4u,sun4m
  creation_node=pumbaa
  creation_hardware_class=sun4u
  creation_platform=SUNW,Sun-Fire
  creation_processor=sparc
  creation_release=5.9
  creation_os_name=SunOS
  creation_os_version=s81_49
  x-department=Internal Finance

▼ To Update a Master Image and Create a Differential Archive

Before creating a differential archive, you need two images to compare: an unchanged master image and an updated master image. One image is the unchanged master image that has been kept unchanged. This image was stored and needs to accessed. The second image is the unchanged master image that is updated with minor changes. The root (/) file system is the default for the new image, but you can access this image if it has been stored elsewhere. After you have the two images, you can create a differential archive, which contains only the differences between the two images. The differential archive can then be installed on clones that were installed previously with the unchanged master image.

1. Prepare the master system with changes. Before changes are made, the master system should be running a duplicate of the original archive.
Note – A copy of the unchanged master image must be kept protected from changes and available for mounting later.

2. Update the unchanged master image with any of the following changes.
   - Delete packages.
   - Add packages or patches.
   - Modify configuration files.
   - Add support for peripheral devices on the clone system.

3. (Optional) Create custom scripts. See “Creating Customization Scripts” on page 171.

4. Deliver the unchanged-master image in a mount point.
   - If the unchanged master image is stored on an inactive boot environment, retrieve by using the `lumount` command.
     
     ```bash
     # lumount BE_name mountpoint
     ```

     | BE_name          | Specifies the boot environment name where the unchanged master image is stored |
     |------------------|--------------------------------------------------------------------------------|
     | mountpoint       | Specifies a root (/) file system where the image is stored                     |

     In the following example, the inactive boot environment is named `unchanged_master1`. The mount point is the directory `/a` on the master system.

     ```bash
     # lumount unchanged_master1 /a
     ```

   - If the image is stored on a clone, mount the clone by using NFS.
     a. On the master system, share the clone’s root (/) file system and give the master root permissions on the clone system.

        ```bash
        # share -F nfs -o rw,root=master_system "/"
        ```

        `master_system` is the name of the master system.

     b. On the master system, mount the clone.

        ```bash
        # mount -F nfs clone_system:/ master_dir
        ```

        | clone_system      | Specifies the name of the system to be mounted                                |
        |-------------------|-------------------------------------------------------------------------------|
        | master_dir        | Specifies the directory where the unchanged master image is stored.           |
If you saved the image with the `ufsdump` command, use the `ufsrestore` command to retrieve a copy. For information on how to use these commands, see “UFS Backup and Restore Commands (Reference)” in System Administration Guide: Basic Administration.

5. Create the differential archive.

```
# flar create -n archive_name -A unchanged_master_image_dir
  options path/filename
```

- **archive_name**
  Specifies the name that you give the archive. The `archive_name` you specify is the value of the `content_name` keyword. The name is listed in the archive identification section.

- **-A unchanged_master_image_dir**
  Creates a differential archive by comparing a new system image with the image that is specified by the `unchanged_master_image_dir` argument. By default, the new system image is root (`/`). You can change the default with the `-R` option. `unchanged_master_image_dir` is a directory where the unchanged system image is stored or mounted through UFS, NFS, or `lumount`.

  You can include and exclude some files by using the options for contents selection. For a list of options, see “`flar create`” on page 198.

- **options**
  For a description of options, see “`flar create`” on page 198.

- **path**
  Specifies the path to the directory in which you want to save the archive file. If you do not specify a path, `flar create` saves the archive file in the current directory.

- **filename**
  Specifies the name of the archive file.

- If the differential archive creation is successful, the `flar create` command returns an exit code of 0.
- If the differential archive creation fails, the `flar create` command returns a nonzero exit code.

**Examples — Creating a Differential Archive**

**EXAMPLE 18-13** Creating a Differential Archive With the New Master Image On the Master System

In this example, the directory for unchanged master image is named `unchanged_master1`. The new master image that contains changes is the root (`/`) directory. The new master image is compared to the unchanged master image and the resulting differential archive is then compressed. The differential archive is stored in `diffarchive1.flar` file. The archive contains files that are to be deleted, changed, or added when installed.

```
# flar create -n diffarchive1 -A /a/unchanged_master1 -c diffarchive1.flar
```
EXAMPLE 18–14 Creating a Differential Archive With the Images Stored on an Inactive Boot Environment

In this example, the unchanged master image, unchanged_master1, is stored on an inactive boot environment and is accessed by mounting the boot environment. The new master image is the root (/) directory. The new master image is compared to the unchanged master and the resulting differential archive is then compressed. The archive is stored in diffarchive4.flar. The archive contains files that are to be deleted, changed, or added when installed.

# lumount unchanged_master1 /a
# flar create -n diffarchive4 -A /a -c diffarchive4.flar
Installing and Administering Solaris Flash Archives (Tasks)

This chapter provides references to procedures for installing Solaris Flash archives on clone systems and procedures for administering an archive.

- “References to Procedures for Installing Solaris Flash Archives” on page 185
- “Administering Solaris Flash Archives” on page 186

References to Procedures for Installing Solaris Flash Archives

You can use any of the Solaris installation methods to install Solaris Flash archives for an initial installation. You must use custom JumpStart to install a Solaris Flash differential archive.

- For an initial installation to install a Solaris Flash archive, see the following references.
  - Solaris Live Upgrade – See “Installing Solaris Flash Archives on a Boot Environment” on page 372
  - The Solaris Web Start program – See “SPARC: Performing an Installation or Upgrade With the Solaris Web Start Program” on page 126 or “x86: Performing an Installation or Upgrade With the Solaris Web Start Program” on page 131
  - The Solaris suninstall program – See “SPARC: Performing an Installation or Upgrade With the Solaris suninstall Program” on page 142 or “x86: Performing an Installation or Upgrade With the Solaris suninstall Program” on page 145
  - Custom JumpStart installation program – See “Creating a Profile” on page 225 and “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 257

- For an update with a Solaris Flash differential archive, see the following references.
Custom JumpStart installation program – See “Creating a Profile” on page 225 and “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 257.

Administering Solaris Flash Archives

The flar command enables you to administer archives. You can split an archive into sections. Those sections can be modified, added to, or deleted, and then merged to create an archive. You can also obtain information about the archive.

Caution – Do not modify the Archive Files section or you compromise the integrity of the archive.

Splitting a Solaris Flash Archive

You can split an archive into sections, which enables you to modify some sections, add new sections, or delete sections. After you have modified the sections, you need to merge the sections to create a new archive. For example, you might want to add a User-Defined section or modify the Archive Identification section. Do not modify the Archive Files section or you compromise the integrity of the archive.

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

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

-\(d\) dir Retrieves the sections to copy from dir, rather than from the current directory
-u section

- If you use this option, flar copies the Cookie, Identification, Archive, and section sections. You can specify a single section name or a space-separated list of section names.
- If you do not use this option, flar copies the Cookie, Identification, and Archive sections only.

-f archive

Extracts the Archive section into a directory that is named archive, rather than placing it in a file with the name archive.

-s section

Only copies the section that is named section from the archive. This section is user-defined.

EXAMPLE 19-1 Splitting an Archive

In the following example, archive1.flar is split into three files:
- cookie – The first line of the archive, which identifies the version of the archive format. Do not change this identifier.
- identification – A copy of the Archive Identification section with all keyword-value pairs.
- archive – The cpio archive itself. This file can be compressed.

# flar split archive1.flar

After the archive is split, you can modify the Archive Identification section or add a User-Defined section. The sections can then be merged to recreate the archive.

Merging a Solaris Flash Archive

After you have split an archive into sections, you can combine the sections to create a new archive.

The flar combine command creates a Solaris Flash archive from individual sections. Each section is assumed to be in a separate file, the names of which are the section names. At a minimum, these three files must be present:
- Archive Cookie (cookie)
- Archive Identification (identification)
- Archive Files (archive)

When combining sections, remember the following points:
- If archive is a directory, flar uses cpio to archive the directory before including it in the combined archive.
- If the Archive Identification section specifies to compress the archive, flar compresses the contents of the newly combined archive.
- No validation is performed on any of the sections. In particular, no fields in the Archive Identification section are validated or updated.

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

- `-d dir` Retrieves the sections to combine from `dir`, rather than from the current directory.
- `-u section` If you use this option, `flar` copies the Cookie, Identification, Archive, and `section` sections. You can specify a single section name or a space-separated list of section names.
  
  If you do not use this option, `flar` copies the Cookie, Identification, and Archive sections only.

---

**EXAMPLE 19-2** Merging a Solaris Flash Archive

In this example, an Archive Cookie section, an Archive Identification section, and an Archive Files section are combined to become a complete archive. The archive is named `newarchive.flar`.

```bash
# flar combine newarchive.flar
```

**EXAMPLE 19-3** Merging a Solaris Flash Archive and Adding a User-Defined Section

In this example, an Archive Cookie section, an Archive Identification section, an Archive Files section, and a User-Defined section are combined to become a complete archive. The archive is named `newarchive.flar`. The User-Defined section content is in the file that is named `user-defined`, which is in the current directory.

```bash
# flar combine -u user_defined newarchive.flar
```

---

**Extracting Information From an Archive**

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

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

- `-k keyword` Returns only the value of the keyword `keyword`.
- `-l` Lists all the files in the archive section.

**EXAMPLE 19-4**

In this example, the file structure of the archive named, `archive3.flar`, is checked.
EXAMPLE 19–4  (Continued)

```
# flar info -l archive3.flar
  aaa
  aaa/bbb
  aaa/bbb/ccc
  aaa/bbb/ccc/ddd
  aaa/eee
```
Solaris Flash (Reference)

This chapter provides a description of Solaris Flash sections, keywords, and keyword values. Also, the chapter describes the `flar create` command options.

- “Solaris Flash Archive Section Descriptions” on page 191
- “Solaris Flash Keywords” on page 193
- “Solaris Flash `flar create` Command” on page 198

Solaris Flash Archive Section Descriptions

Each Solaris Flash archive is grouped into sections. Some sections are generated by the Solaris Flash software and need no input from you. Some sections require input or optionally allow you to add information. The following table describes each section.

<table>
<thead>
<tr>
<th>Section Name</th>
<th>Description</th>
<th>Required by Archive?</th>
<th>Requires Input From User?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive cookie</td>
<td>The first section contains a cookie that identifies the file as a Solaris Flash archive. The deployment code uses the cookie for identification and validation purposes. The cookie must be present for an archive to be valid.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Section Name</td>
<td>Description</td>
<td>Required by Archive?</td>
<td>Requires Input From User?</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>---------------------------</td>
</tr>
</tbody>
</table>
| Archive identification | The second section contains keywords with values that provide identification information about the archive. The software generates some information such as the following:  
- The archive ID number  
- The method of archival such as cpio  
- The creation date by default  
You are required to specify a name for the Solaris Flash archive. Other information that you can specify about the archive includes:  
- The author of the archive  
- The date that the archive was created  
- The name of the master system that you used to create the archive  
For a list of keywords that describe the archive, see “Identification Section Keywords” on page 194 | Yes                  | Content is generated by both user and the software |
| Manifest         | A section of a Solaris Flash archive that is used to validate a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. The installation fails if the files do not match the expected file set. This section is informational only. The section lists the files in an internal format and cannot be used for scripting.  
You can exclude this section by creating the differential archive with the flar create -M option. Because no validation of the archive occurs, excluding this section is not recommended. | No                   | No                        |
|                  | Predeployment  
|                  | Postdeployment  
<p>|                  | Reboot          | This section contains internal information that the flash software uses before and after installing an operating environment image. Any customization scripts that you have provided are stored in this section. | Yes                  | No                        |
| Summary          | This section contains messages about the archive creation and records the activities of predeployment scripts.                                                                                               | Yes                  | Content is generated by both user and the software |
| User-defined     | This section follows the identification section. The archive can contain zero or more user-defined sections. These sections are not processed by thearchive extraction code. These sections are retrieved separately and can be used for content descriptions. | No                   | Yes                       |</p>
<table>
<thead>
<tr>
<th>Section Name</th>
<th>Description</th>
<th>Required by Archive?</th>
<th>Requires Input From User?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive files</td>
<td>The archive files section contains the files that have been gathered from the master system in binary date. This section begins with section_begin=archive, but it does not have an ending section boundary.</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**Solaris Flash Keywords**

Solaris Flash keywords are like custom JumpStart keywords. They define elements of the installation. Each keyword is a command that controls one aspect of how the Solaris Flash software installs the software on a clone system.

Use the following guidelines to format keywords and values:

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

**General Keywords**

Each Solaris Flash archive section is defined by the `section_begin` and `section_end` keywords. For example, the archive files section includes a `section_begin` keyword, though with a different value. User-defined archive sections are delimited by `section_begin` and `section_end` keywords, with values appropriate to each section. The values for the `section_begin` and `section_end` keywords are described in the following table.

**TABLE 20-2 Values for section_begin and section_end Keywords**

<table>
<thead>
<tr>
<th>Archive Section</th>
<th>Value for section_begin and section_end keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive cookie</td>
<td>cookie — This section is not delimited by the section_begin and section_end keywords.</td>
</tr>
<tr>
<td>Archive identification</td>
<td>identification</td>
</tr>
<tr>
<td>User-defined sections</td>
<td>section_name An example of a section_name keyword is X-user_section_1.</td>
</tr>
</tbody>
</table>
TABLE 20–2 Values for section_begin and section_end Keywords  (Continued)

<table>
<thead>
<tr>
<th>Archive Section</th>
<th>Value for section_begin and section_end keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive files</td>
<td>archive</td>
</tr>
</tbody>
</table>

Identification Section Keywords

The following tables describe the keywords for use in the Archive Identification section and the values you can define for them.

Every section uses the keywords in Table 20–3 to delimit each section.

TABLE 20–3 Identification Section Keywords: General Keywords

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Value Definitions</th>
<th>Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>section_begin</td>
<td>These keywords are used to delimit sections in the archive and are not limited</td>
<td>Text</td>
<td>Yes</td>
</tr>
<tr>
<td>section_end</td>
<td>exclusively to the identification section. For a description of these keywords,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>see “General Keywords” on page 193.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following keywords, used in the archive identification section, describe the contents of the archive files section.

TABLE 20–4 Identification Section Keywords: Contents of Archive Files Section

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Value Definitions</th>
<th>Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>archive_id (optional)</td>
<td>This keyword uniquely describes the contents of the archive. This value is used</td>
<td>Text</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>by the installation software only to validate the contents of the archive during</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>archive installation. If the keyword is not present, no integrity check is</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>performed. For example, the archive_id keyword might be FlAsH-ARCHive-2.0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>files_archived_method</td>
<td>This keyword describes the archive method that is used in the files section.</td>
<td>Text</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>▪ If the keyword is present, it has the value of cpio</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ If this keyword is not present, the files section is assumed to be in CPIO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>format with ASCII headers. This format is the cpio -c option.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the files_compressed_method is present, the compression method is applied to</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the archive file that is created by the archive method.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 20–4 Identification Section Keywords: Contents of Archive Files Section

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Value Definitions</th>
<th>Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>files_archived_size</code></td>
<td>This keyword value is the size of the archived files section in bytes.</td>
<td>Numeric</td>
<td>No</td>
</tr>
<tr>
<td><code>files_compress_method</code></td>
<td>This keyword describes the compression algorithm that is used on the files section.</td>
<td>Text</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>- If the keyword is present, it can have one of the following values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- none – The archive file section is not compressed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- compress – The file section is compressed by using the <code>compress</code> command</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- If this keyword is not present, the archive files section is assumed to be uncompressed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The compression method indicated by this keyword is applied to the archive file created by the archive method indicated by the <code>files_archived_method</code> keyword.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>files_unarchived_size</code></td>
<td>This keyword defines the cumulative size in bytes of the extracted archive. The value is used for file-system size verification.</td>
<td>Numeric</td>
<td>No</td>
</tr>
</tbody>
</table>

The following keywords provide descriptive information about the entire archive. These keywords are generally used to assist you in archive selection and to aid in archive management. These keywords are all optional and are used to help you to distinguish between individual archives. You use options for the `flar create` command to include these keywords. For an example, see Example 18–12.

### TABLE 20–5 Identification Section Keywords: User Describes the Archive

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Value Definitions</th>
<th>Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>creation_date</code></td>
<td>This keyword value is a textual timestamp that represents the time that you created the archive.</td>
<td>Text</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>- You can use <code>flar create</code> command with the <code>-i</code> option to create the date.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- If you do not specify a creation date with the <code>flar create</code> command, the default date is set in Greenwich mean time (GMT).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- The value must be in ISO-8601 complete basic calendar format without the time designator (ISO-8601,§5.4.1(a)). The format is <code>CCYYMMDDhhmmss</code>. For example, 20000131221409 represents January 31st, 2000 10:14:09 p.m.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keywords</td>
<td>Value Definitions</td>
<td>Value</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>creation_master</td>
<td>This keyword value is the name of the master system you used to create the archive.</td>
<td>Text</td>
<td>No</td>
</tr>
</tbody>
</table>
| content_name         | This keyword identifies the archive. The value is generated from the flar create -n option. Follow these guidelines when you create this value:  
  - The descriptive name can be no longer than 256 characters.
  - The description should contain the function and purpose of the archive. | Text     | Yes      |
| content_type         | This keyword value specifies a category for the archive. You use the flar create -T option to generate the value. | Text     | No       |
| content_description  | The keyword value describes the contents of the archive. The value of this keyword has no length limit. You use the flar create -E option to create this value. | Text     | No       |
| content_author       | This keyword value identifies the creator of the archive. You use the flar create-a option to create this value. Suggested values include the full name of the creator and the creator’s email address. | Text     | No       |
| content_architectures| This keyword value is a comma-separated list of the kernel architectures that the archive supports. You can use the flar create ? option to create this value.  
  - If the keyword is present, the installation software validates the kernel architecture of the clone system against the list of architectures that the archive supports. The installation fails if the archive does not support the kernel architecture of the clone system.  
  - If the keyword is not present, the installation software does not validate the architecture of the clone system. | Text list | No       |

The following keywords also describe the entire archive. By default, the values are filled in by uname when the flash archive is created. If you create a flash archive in which the root directory is not /, the archive software inserts the string UNKNOWN for the keywords. The exceptions are the creation_node, creation_release, and creation_os_name keywords.
For `creation_node`, the software uses the contents of the `nodename` file.

For `creation_release` and `creation_os_name`, the software attempts to use the contents of root directory `/var/sadm/system/admin/INST_RELEASE`. If the software is unsuccessful in reading this file, it assigns the value `UNKNOWN`.

Regardless of their sources, you cannot override the values of these keywords.

### TABLE 20-6 Identification Section Keywords: Software Describes the Archive

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>creation_node</code></td>
<td>The return from <code>uname -n</code></td>
</tr>
<tr>
<td><code>creation_hardware_class</code></td>
<td>The return from <code>uname -m</code></td>
</tr>
<tr>
<td><code>creation_platform</code></td>
<td>The return from <code>uname -i</code></td>
</tr>
<tr>
<td><code>creation_processor</code></td>
<td>The return from <code>uname -p</code></td>
</tr>
<tr>
<td><code>creation_release</code></td>
<td>The return from <code>uname -r</code></td>
</tr>
<tr>
<td><code>creation_os_name</code></td>
<td>The return from <code>uname -s</code></td>
</tr>
<tr>
<td><code>creation_os_version</code></td>
<td>The return from <code>uname -v</code></td>
</tr>
</tbody>
</table>

### User-Defined Section Keywords

In addition to the keywords that are defined by the Solaris Flash archive, you can define other keywords. The Solaris Flash archive ignores user-defined keywords, but you can provide scripts or programs that process the identification section and use user-defined keywords. Use the following format when creating user-defined keywords:

- Begin the keyword name with an `X`.
- Create the keyword with any characters other than linefeeds, equal signs, and null characters.
- Suggested naming conventions for user-defined keywords include the underscore-delimited descriptive method used for the predefined keywords. Another convention is a federated convention similar to the naming of Java packages.

For example, `X-department` is a valid name for a user-defined keyword.

For an example of using options to include user-defined keywords in the identification section, see Example 18–12.
Solaris Flash flar create Command

Use the Solaris Flash flar create command to create a Solaris Flash archive.

flar create

Use the flar create command to create a Solaris Flash archive from a master system. You can use this command when the master system is running in multiuser mode or single-user mode. You can also use flar create when the master system is booted from the Solaris 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 Solaris Flash archive. The syntax of the command is as follows:


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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required Options</strong></td>
<td></td>
</tr>
<tr>
<td>-n archive_name</td>
<td>The value of this flag is the name of the archive. The archive_name you specify is the value of the content_name keyword.</td>
</tr>
<tr>
<td><strong>Option for Compression</strong></td>
<td></td>
</tr>
<tr>
<td>-c</td>
<td>Compresses the archive by using compress(1).</td>
</tr>
<tr>
<td><strong>Options for Directories and Sizes</strong></td>
<td></td>
</tr>
<tr>
<td>-R root</td>
<td>Creates the archive from the file system tree that is rooted at root. If you do not specify this option, flar create creates an archive from a file system that is rooted at /.</td>
</tr>
<tr>
<td>-S</td>
<td>Omits sizing information in the archive.</td>
</tr>
</tbody>
</table>
### TABLE 20–7 Command-Line Options for flarc create  (Continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-H</code></td>
<td>Does not generate the hash identifier.</td>
</tr>
</tbody>
</table>

#### Options for Creating a Differential Archive

- `-A unchanged_master_image_dir`  
  Create a differential archive by comparing a new system image with the image that is specified by the `unchanged_master_image_dir` argument. By default, the new system image is root (`/`). You can change the default with the `-R` option. `unchanged_master_image_dir` is a directory where the unchanged master system image is stored or mounted through UFS, NFS, or `lumount`.

  You can modify the effects of file selection for a differential archive by using the options for contents selection described in the next section of the table.

- `-M`  
  Excludes the manifest file. When you use this option, no validation occurs on the differential archive. When creating a differential archive, flarc create creates a long list of the files in the system that are unchanged, are changed, and are to be deleted from the archive. This list is stored in the manifest section of the archive. When the differential archive is deployed, the software uses this list to perform a file-by-file check, ensuring the integrity of the clone system. Use of this option avoids such a check and saves the space that is used by the manifest section in a differential archive. However, you must consider the savings in time and disk space against the loss of an integrity check upon installation. Because no validation occurs, avoid using this option.

#### Options for Contents Selection

**Caution** – Use the flarc create file-exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.

- `-y include_dir/filename`  
  Adds to the archive those files and directories that are specified on the command line. This option is used when you’ve excluded a directory, but want to restore individual subdirectories or files.

  *include_dir/filename* is the name of the subdirectory or file to be included.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| -f list_filename | Adds files and directories from a list to the archive. list_filename is the full path to a file that contains a list. The contents of the file are added to the file list unless -F is specified.  
  - The list_filename file must contain one file per line.  
  - If you specify a file system with -R root, the path to each file must be relative to the alternate root directory or an absolute path.  
  - If file name is "-", flar create reads standard input as the list of files. When you use the value "-", the archive size is not calculated. |
| -F             | Uses only the files in -f list_filename to create the archive. This option makes the -f list_filename the absolute list, rather than a list that is appended to the normal file list. |
| -x exclude_dir/filename | Excludes files and directories from the archive. These files and directories are specified at the command line. You can use multiple instances of this option to exclude more than one file or directory. exclude_dir/filename is the name of the directory or file to be excluded. |
| -X list_filename | Excludes a list of files or directories from the archive. list_filename is the full path to a file that contains the list.  
  - The list_filename file must contain one file per line.  
  - If you specify a file system with -R root, the path to each file must be relative to the alternate root directory or an absolute path.  
  - If list_filename is "-", flar create reads standard input as the list of files. When you use the value "-", the archive size is not calculated. |
| -z list_filename | Excludes or includes a list of files or directories from the archive. Each file or directory in the list is noted with a plus "+" or minus "-". A plus indicates an included file or directory and the minus indicates an excluded file or directory. list_filename is the full path to a file that contains the list.  
  - The list_filename file must contain one file per line.  
  - If you specify a file system with -R root, the path to each file must be relative to the alternate root directory or an absolute path. |
TABLE 20–7 Command-Line Options for flar create  (Continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-I</td>
<td>Override the integrity check. To prevent you from excluding important system files from an archive, flar create runs an integrity check. This check examines all files that are registered in a system package database and stops archive creation if any of them are excluded. Use of this option overrides this integrity check. Therefore, avoid the use of the -I option.</td>
</tr>
</tbody>
</table>

Options Used With User-Defined Sections

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-u section</td>
<td>Includes section as a user-defined section. To include more than one user-defined section, section must be a space-separated list of section names.</td>
</tr>
<tr>
<td>-d dir</td>
<td>Retrieves the section file that is specified with -u from dir.</td>
</tr>
</tbody>
</table>

Options Used With Tape Archives

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-t</td>
<td>Creates an archive on a tape device. The filename argument is the name of the tape device.</td>
</tr>
<tr>
<td>-p posn</td>
<td>Use only with the -t option. Specifies the position on the tape device for flar create to store the archive. If you do not use this option, flar create places the archive at the current position of the tape.</td>
</tr>
<tr>
<td>-b blocksize</td>
<td>Specifies the block size flar create uses when creating the archive. If you do not specify a block size, flar create uses the default block size of 64k.</td>
</tr>
</tbody>
</table>

Options for Archive Identification

These keywords and values appear in the identification section of the archive.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-U key=val</td>
<td>Includes user-defined keywords and values in the Archive Identification section.</td>
</tr>
<tr>
<td>-i date</td>
<td>Uses date as the value for the creation_date keyword. If you do not specify a date, flar create uses the current system time and date.</td>
</tr>
<tr>
<td>-m master</td>
<td>Uses master as the name of the master system on which you created the archive. master is the value for the creation_master keyword. If you do not specify master, flar create uses the system name that is reported by uname -n.</td>
</tr>
<tr>
<td>-e descr</td>
<td>Uses descr for the value of the content_description keyword. You cannot use this option when you use the -E option.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>-E descr_file</code></td>
<td>Retrieves the value for the <code>content_description</code> keyword from the <code>descr_file</code> file. You cannot use this option when you use the <code>-e</code> option.</td>
</tr>
<tr>
<td><code>-a author</code></td>
<td>Uses <code>author</code> as the author name in the Archive Identification section. <code>author</code> is the value for the <code>content_author</code> keyword. If you do not specify an author, <code>flar create</code> does not include the <code>content_author</code> keyword in the Archive Identification section.</td>
</tr>
<tr>
<td><code>-T type</code></td>
<td>Uses <code>type</code> as the value for the <code>content_type</code> keyword. <code>type</code> is user defined. If you do not specify a type, <code>flar create</code> does not include the <code>content_type</code> keyword.</td>
</tr>
</tbody>
</table>
Custom JumpStart Installation (Topics)

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

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 22</td>
<td>Provides an introduction and overview of the custom JumpStart installation method.</td>
</tr>
<tr>
<td>Chapter 23</td>
<td>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.</td>
</tr>
<tr>
<td>Chapter 24</td>
<td>Describes the optional features that you can use to create additional tools for a custom JumpStart installation.</td>
</tr>
<tr>
<td>Chapter 25</td>
<td>Provides information and procedures for creating your own custom rule and custom probe keywords.</td>
</tr>
<tr>
<td>Chapter 26</td>
<td>Describes how to perform a custom JumpStart installation on a SPARC based or an x86 based system. You need to follow these procedures on the system on which you intend to install the Solaris 9 software.</td>
</tr>
<tr>
<td>Chapter 27</td>
<td>Provides an example of setting up and installing Solaris software on both SPARC based and x86 based systems by using the custom JumpStart installation method.</td>
</tr>
<tr>
<td>Chapter 28</td>
<td>Contains lists of keywords and values to be used in the rules file, profiles, begin scripts, and finish scripts.</td>
</tr>
</tbody>
</table>
Custom JumpStart (Overview)

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

- “Custom JumpStart Introduction” on page 205
- “How the JumpStart Program Installs Solaris Software” on page 206

Custom JumpStart Introduction

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

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

- Install Solaris on 100 new systems.
- Seventy of the systems are SPARC systems that are owned by the engineering group and need to be installed as standalone systems with the Solaris operating environment software group for developers.
- The remaining 30 systems are x86 based, owned by the marketing group and need to be installed as standalone systems with the Solaris operating environment software group for end users.
First, the system administrator must create a rules file and a profile for each group of systems. The rules file is a text file that contains a rule for each group of systems or single systems on which you want to install the Solaris software. Each rule distinguishes a group of systems that are based on one or more system attributes. Each rule also links each group to a profile.

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

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

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

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

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

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

How the JumpStart Program Installs Solaris Software

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

Does the system find a diskette in the system’s diskette drive?

Yes

The system mounts the diskette.

Yes

Does the system find a rules.ok file on the diskette?

No

Does the system find a rules.ok file on a designated server?

Yes

Does the system match any rules in the rules.ok file?

No

END

The system proceeds with an alternate installation program.

No

END

The system proceeds with a custom JumpStart installation. The profile specified in the matched rule is used to install the system.

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 211
- “Creating a Profile Server for Networked Systems” on page 213
- “Creating a Profile Diskette for Standalone Systems” on page 217
- “Creating the rules File” on page 221
- “Creating a Profile” on page 225
- “Testing a Profile” on page 229
- “Validating the rules File” on page 233

### Task Map: Preparing Custom JumpStart Installations

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decide how to upgrade the system if a previous version of the Solaris software is installed on the system</td>
<td>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.</td>
<td>Chapter 8</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
<td>For Instructions</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Create a JumpStart directory  | **On a server**  
If you want to perform custom JumpStart installations on systems that are connected to a network, you must create a profile server. The profile server contains a JumpStart directory for the custom JumpStart files.  
**On a diskette**  
If you want to perform custom JumpStart installations on systems that are not connected to a network, you must create a profile diskette. A profile diskette contains the custom JumpStart files. | “Creating a Profile Server for Networked Systems” on page 213  
“Creating a Profile Diskette for Standalone Systems” on page 217 |
| Add rules to the rules file   | After you decide how you want each group of systems or single systems to be installed, create a rule for each group that you want to install. Each rule distinguishes a group, based on one or more system attributes. The rule links each group to a profile. | “Creating the rules File” on page 221 |
| Create a profile for every rule | A profile is a text file that defines how to install the Solaris software, for example, which software group to install on a system. Every rule specifies a profile to define how a system is to be installed with the Solaris software when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule. | “Creating a Profile” on page 225 |
| (Optional) Test the profiles  | After you create a profile, use the pfinstall(1M) command to test the profile before you use the profile to install or upgrade a system. | “Testing a Profile” on page 229 |
### Creating a Profile Server for Networked Systems

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

The server that contains a JumpStart directory is called a profile server. A profile server can be the same system as an install server or a boot server, or the server can be a completely different server. A profile server can provide custom JumpStart files for different platforms. For example, an x86 server can provide custom JumpStart files for both SPARC systems and x86 systems.

**Note** – After you create a profile server, you must allow systems to access the server. For detailed instructions, see “To Allow All Systems Access to the Profile Server” on page 215.

#### 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.

```bash
# 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:

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

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

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

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

```bash
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.

<table>
<thead>
<tr>
<th>Example Locations</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD for your platform</td>
<td>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.</td>
</tr>
<tr>
<td>An image of the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD for your platform on a local disk</td>
<td>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></td>
</tr>
</tbody>
</table>

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

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

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

For example, the following command copies the `jumpstart_sample` directory into the `/jumpstart` directory on the profile server:
For SPARC systems:
```
cp -r /cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample/* /jumpstart
```

For x86 systems:
```
cp -r /cdrom/cdrom0/s2/Solaris_9/Misc/jumpstart_sample/* /jumpstart
```

7. Update the example JumpStart files so that the files work in your environment.

8. Ensure that `root` owns the JumpStart directory and that the permissions are set to 755.

9. Allow systems on the network to access the profile server.
   For detailed instructions, see “To Allow All Systems Access to the Profile Server” on page 215.

▼ To Allow All Systems Access to the Profile Server

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

- `add_install_client` command – Each time that you add a system for network installation, use the `-c` option with the `add_install_client` command. For detailed instructions, refer to “Adding Systems to Be Installed From the Network With a CD Image” on page 119.

- `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 242.

When you boot the system to initiate the custom JumpStart installation, specify the location of the compressed file. For detailed instructions for SPARC systems, refer to Step 5 in “SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 258. For x86 systems refer to Step 8 in “x86: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 262.

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

---

**Note** – The following procedure is not necessary if you save the JumpStart directory on a diskette or if you specify the location of the profile server when you boot the system.
The following procedure is valid only if you store network installation information in the /etc/bootparams file. You can also store network installation information in one of the following places:

- Name service database – If you store network installation information in the name service bootparams database, you must update the bootparams database with the entry that is shown in Step 3.
- DHCP server – If you store network installation information on a DHCP server, use the boot command to specify that the custom JumpStart program use the DHCP server. For detailed instructions for SPARC systems, refer to Step 5 in “SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 258. For x86 systems, refer to Step 8 in “x86: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 262.

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
   ```

   * `install_config` The wildcard character that specifies that all systems have access
   * `server` The host name of the profile server where the JumpStart directory is located
   * `jumpstart_dir_path` The absolute path of the JumpStart directory

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

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

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

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

   “Booting From the Network, Error Messages” on page 445 contains details about this error message.

   All systems can now access the profile server.
Creating a Profile Diskette for Standalone Systems

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

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

▼ SPARC: To Create a Profile Diskette

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

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

<table>
<thead>
<tr>
<th>Example Locations</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>An image of the Solaris 9 SPARC Platform Edition DVD or the Solaris 9 Software 1 of 2 SPARC Platform Edition CD on a local disk</td>
<td>Change the directory to the location of the Solaris 9 SPARC Platform Edition DVD or the Solaris 9 Software 1 of 2 SPARC Platform Edition CD image. For example, type the following command: <code>cd /export/install</code></td>
</tr>
</tbody>
</table>

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

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

- `media_path`: The path to the CD, DVD, or image on the local disk
- `jumpstart_dir_path`: The path to the profile diskette where you want to place the example custom JumpStart files

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

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

```
  cp -r /cdrom/sol_9_sparc/s0/Solaris_9/Misc/jumpstart_sample/* /floppy/scrap
```
9. Update the example JumpStart files on the profile diskette so that the files work in your environment.

10. Ensure that root owns the JumpStart directory and that permissions are set to 755.

11. Eject the diskette.

```
# eject floppy
```

You have completed the creation of a profile diskette. You can now update the rules file and create profiles on the profile diskette to perform custom JumpStart installations. To continue, go to “Creating the rules File” on page 221.

▼ x86: To Create a Profile Diskette

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

1. Log in as superuser on an x86 system to which a diskette drive is attached.

2. Insert the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette into the diskette drive (usually drive A). You use this diskette as the profile diskette.

**x86 only** – You can access the boot diskette software by downloading and copying the software to a diskette from the Solaris Developer Connection at [http://soldc.sun.com/support/drivers/dcs_diskettes](http://soldc.sun.com/support/drivers/dcs_diskettes).

3. Mount the diskette.

```
# volcheck
```

4. Copy the image of the Solaris 9 Device Configuration Assistant to the system’s hard disk.

```
# dd if=/vol/dev/aliases/floppy0 of=boot_image
```

In the command, *boot_image* is the name of the file into which you want to copy the image of the Solaris 9 Device Configuration Assistant. You can specify an absolute path name.

For example, the following command copies the boot diskette to a file that is named boot_save:

```
dd if=/vol/dev/aliases/floppy0 of=boot_save
```
5. Eject the diskette by clicking Eject Disk in the File Manager window or by typing `eject floppy` on the command line.

6. In the Removable Media Manager dialog box, click OK.


8. Insert a blank diskette or a diskette that can be overwritten into the diskette drive.

9. Mount the diskette.
   
   ```
   # volcheck
   ```

10. Format the diskette.

    ! Caution – Formatting erases all data on the diskette.

    ```
    # fdformat -d -U
    ```

11. Copy the Solaris 9 Device Configuration Assistant image from the system’s hard disk to the formatted diskette.

    ```
    # dd if=boot_image of=/vol/dev/aliases/floppy0
    ```

    In the command, `boot_image` is the name of the file where you want to copy the image of the Solaris 9 Device Configuration Assistant. You can specify an absolute path name.

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

    - If no, go to Step 15
    - If yes, use the following decision table to determine what to do next.

<table>
<thead>
<tr>
<th>Example Locations</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 1 of 2 x86 Platform Edition CD</td>
<td>Insert the Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 1 of 2 x86 Platform Edition CD into the server’s CD-ROM drive. Volume Manager automatically mounts the CD.</td>
</tr>
<tr>
<td>An image of the Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 1 of 2 x86 Platform Edition CD on a local disk</td>
<td>Change directory to the location of the Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 1 of 2 x86 Platform Edition CD image. For example, type the following: cd/export/install</td>
</tr>
</tbody>
</table>
13. Copy the example custom JumpStart files into the JumpStart directory on the profile diskette.

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

<table>
<thead>
<tr>
<th>media_path</th>
<th>The path to the CD, DVD, or image on the local disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>jumpstart_dir_path</td>
<td>The path to the profile diskette where you want to place the example custom JumpStart files</td>
</tr>
</tbody>
</table>

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

For example, the following command copies the contents of `jumpstart_sample` on the Solaris 9 Software 1 of 2 x86 Platform Edition CD to the root (/) directory on a profile diskette that is named `scrap`:

```bash
cp -r /cdrom/sol_9_x86/s2/Solaris_9/Misc/jumpstart_sample/* /floppy/scrap
```

14. Update the example JumpStart files on the profile diskette so that the files work in your environment.

15. Ensure that `root` owns the JumpStart directory and that permissions are set to 755.

16. Eject the diskette by clicking Eject Disk in the File Manager window or by typing `eject floppy` on the command line.

17. In the Removable Media Manager dialog box, click OK.

18. Manually eject the diskette.

You have completed the creation of a profile diskette. Now you can update the `rules` file and create profiles on the profile diskette to perform custom JumpStart installations. To continue, go to “Creating the rules File” on page 221.

---

### 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 217 or “Creating a Profile Server for Networked Systems” on page 213, 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 named 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 277.
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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>!</code></td>
<td>A symbol that is used before a keyword to indicate negation.</td>
</tr>
<tr>
<td><code>rule_keyword</code></td>
<td>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>. <code>rule_keyword</code> 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 277.</td>
</tr>
<tr>
<td><code>rule_value</code></td>
<td>A value that provides the specific system attribute for the corresponding <code>rule_keyword</code>. Rule values are described in “Rule Keywords and Values” on page 277.</td>
</tr>
<tr>
<td><code>&amp;&amp;</code></td>
<td>A symbol you must use to join <code>rule_keyword</code> and <code>rule_value</code> 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.</td>
</tr>
<tr>
<td><code>begin</code></td>
<td>The name of an optional Bourne shell script that can be executed before the installation begins. If no begin script exists, you must type a minus sign (<code>-</code>) in this field. All begin scripts must be located in the JumpStart directory.</td>
</tr>
<tr>
<td><code>profile</code></td>
<td>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. <strong>Note</strong> – Optional ways to use the <code>profile</code> field are described in “Using a Site-Specific Installation Program” on page 248 and “Creating Derived Profiles With a Begin Script” on page 236.</td>
</tr>
<tr>
<td><code>finish</code></td>
<td>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 (<code>-</code>) in this field. All finish scripts must be located in the JumpStart directory.</td>
</tr>
</tbody>
</table>

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-1³ - 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 i86pc setup⁴ x86_prof done
memsize 64-128 & arch i386⁵ - 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” on page 222.

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 an x86 based system.
   The setup begin script, the x86_prof profile, and the done finish script are used to
   install the Solaris software on systems that match the rule.

5. The rule matches if the system has between 64 and 128 Mbytes of memory and is
   an x86 based system. The prog_prof profile is used to install the Solaris software
   on systems that match the rule.

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 213 or “Creating a Profile Diskette for Standalone Systems” on page 217, 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 282.

   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 229 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 282 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_install
system_type standalone
partitioning default
filesys any 512 swap # specify size of /swap
cluster SUNWCprogs
package SUNWman delete
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, SUNWprog, 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 explicit
filesys c0t0d0s0 auto /
filesys c0t3d0s1 auto swap
filesys any auto usr
cluster SUNWCall
```

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 x86: Using the fdisk Keyword

```
# profile keywords profile values
# ---------------- -------------------
install_type initial_install
system_type standalone

fdisk c0t0d0 0x04 delete
fdisk c0t0d0 solaris maxfree
cluster SUNWCall
cluster SUNWCacc delete
```

1. All fdisk partitions of type DOSOS16 (04 hexadecimal) are deleted from the c0t0d0 disk.

2. A Solaris fdisk partition is created on the largest contiguous free space on the c0t0d0 disk.

3. The Entire Distribution software group, SUNWcall, is installed on the system.

4. The system accounting utilities, SUNWCacc, are not to be installed on the system.

EXAMPLE 23-5 Reallocation Disk Space for an Upgrade

```
# profile keywords profile values
# ---------------- -------------------
install_type upgrade
```

Chapter 23 • Preparing Custom JumpStart Installations (Tasks)
EXAMPLE 23–5 Reallocating Disk Space for an Upgrade (Continued)

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. For more backup-media keyword values, see “backup_media Profile Keyword” on page 288.

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.

EXAMPLE 23–6 Retrieving a Solaris Flash Archive From an HTTP Server

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

1. The profile installs a Solaris Flash archive on the clone system. All files are overwritten as in an initial installation.

2. The Solaris Flash archive is retrieved from an HTTP server.
**EXAMPLE 23–6** Retrieving a Solaris Flash Archive From an HTTP Server  
(Continued)

3. The file system slices are determined by the `filesys` keywords, `value explicit`. The size of root (`/`) is based on the selected software and is installed on `c0t1d0s0`. The size of `swap` is set to the necessary size and is installed on `c0t1d0s1`. `/export/home` is based on the selected software and is installed on `c0t1d0s7`.

**EXAMPLE 23–7** Retrieving a Solaris Flash Archive From an NFS Server

In the following example, the profile indicates that the custom JumpStart program retrieves the Solaris Flash archive from an NFS server. The `flash_update` keyword indicates that this is a differential archive. A differential archive installs only the differences between two system images.

```
install_type  flash_update
archive_location nfs installserver:/export/solaris/flasharchive/solaris9archive
no_master_check
root_device c0t0d0s6
```

1. The profile installs a Solaris Flash differential archive on the clone system. Only files that are specified by the archive are installed.
2. The Solaris Flash archive is retrieved from an NFS server.
3. The clone system is not checked for a valid system image. A valid system image would have been built from the original master system.
4. Specifies that the root file system on `c0t0d0s6` is to be updated.

---

**Testing a Profile**

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

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

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

- The system’s disk configuration where `pfinstall` is being run.
- Other disk configurations. You use a disk configuration file that represents a structure of a disk, for example, a disk’s bytes/sector, flags, and slices. Creating disk configuration files is described in “Creating Disk Configuration Files” on page 243 and “x86: To Create a Disk Configuration File” on page 245.
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 SPARC Platform Edition DVD
   - Solaris 9 x86 Platform Edition DVD
   - Solaris 9 Software 1 of 2 SPARC Platform Edition CD
   - Solaris 9 Software 1 of 2 x86 Platform Edition CD

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

2. **Respond to the system identification questions.**

3. **Exit from the installation program.**

   - If you are using the Solaris 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 231.
To Test a Profile

1. Locate a system on which to test the profile that is the same type of platform, SPARC or x86, for which the profile was created.
   If you are testing an upgrade profile, you must test the profile on the actual system that you intend to upgrade.

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

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test an initial installation profile and have a system that is running the Solaris 9 software</td>
<td>Become superuser on the system and go to Step 5.</td>
</tr>
</tbody>
</table>
| 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 230. 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.

<table>
<thead>
<tr>
<th>Mount Scenario</th>
<th>Typing Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mount a remote NFS file system for systems on the network</td>
<td>mount -F nfs server_name:path /tmp/mnt</td>
</tr>
<tr>
<td>SPARC: Mount a UFS-formatted diskette</td>
<td>mount -F ufs /dev/diskette /tmp/mnt</td>
</tr>
<tr>
<td>Mount a PCFS-formatted diskette</td>
<td>mount -F pcfs /dev/diskette /tmp/mnt</td>
</tr>
</tbody>
</table>

5. To test the profile with a specific system memory size, set SYS_MEMSIZE to the specific memory size in Mbytes.
   # SYS_MEMSIZE=memory_size
   # export SYS_MEMSIZE

6. Did you mount a directory in Step 4?
   - If yes, change the directory to /tmp/mnt.
     # cd /tmp/mnt
   - If no, change the directory to where the profile is located, which is usually the JumpStart directory.
     # cd jumpstart_dir_path
7. Test the profile with the `pfinstall(1M)` command.

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

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

- **-D**
  * `pfinstall` uses the current system’s disk configuration to test the profile. You must use the `-D` option to test an upgrade profile.

- **-d disk_config_file**
  * `pfinstall` uses the disk configuration file, `disk_config_file`, to test the profile. If `disk_config_file` is not located in the directory where `pfinstall` is run, you must specify the path.

  For instructions on how to create a disk configuration file, see “Creating Disk Configuration Files” on page 243.

  **Note** – You cannot use the `-d disk_config_file` option with an upgrade profile, `install_type upgrade`. You must always test an upgrade profile against a system’s disk configuration, that is, you must use the `-D` option.

- **-c path**
  * The path to the Solaris 9 software image. You use this option, for example, if the system is using Volume Manager to mount the Solaris 9 Software 1 of 2 CD for your platform.

  **Note** – The `-c` 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 `/dev/cdrom` as part of the booting process.

- **profile**
  * The name of the profile to test. If `profile` is not in the directory where `pfinstall` is being run, you must specify the path.

**Profile Test Examples**

The following example shows how to use `pfinstall` to test a profile that is named `basic_prof`. The profile is tested against the disk configuration on a system on which the Solaris 9 software is installed. The `basic_prof` profile is located in the `/jumpstart` directory, and the path to the Solaris 9 Software 1 of 2 `SPARC Platform Edition` CD or Solaris 9 Software 1 of 2 `x86 Platform Edition` CD image is specified because Volume Manager is being used.

**EXAMPLE 23-8 Profile Test Using a Solaris 9 System**

```
# cd /jumpstart
# /usr/sbin/install.d/pfinstall -D -c /cdrom/pathname basic_prof
```
The following example shows how to use `pfinstall` to test the profile that is named `basic_prof` on a Solaris 9 system. The test is performed against the `535_test` disk configuration file. The test checks for 64 Mbytes of system memory. This example uses a Solaris 9 Software 1 of 2 `SPARC Platform Edition` CD or Solaris 9 Software 1 of 2 `x86 Platform Edition` CD image that is located in the `/export/install` directory.

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

▼ 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:

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

   `-p path` Validates the rules by using the check script from the Solaris 9 software image instead of the check script from the system you are using. `path` is the image on a local disk or a mounted Solaris 9 DVD or a 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 rules file other than the one that is named `rules`. Using this option, you can test the validity of a rule before you integrate the rule into the `rules` file.

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

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

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

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 235
- “Creating Finish Scripts” on page 237
- “Creating a Compressed Configuration File” on page 242
- “Creating Disk Configuration Files” on page 243
- “Using a Site-Specific Installation Program” on page 248

Note — Instructions in this chapter are valid for either a SPARC server or an x86 server that is being used to provide custom JumpStart files, called a profile server. A profile server can provide custom JumpStart files for different platform types. For example, a SPARC server can provide custom JumpStart files for both SPARC systems and x86 systems.

Creating Begin Scripts

A begin script is a user-defined Bourne shell script that you specify in the rules file. A begin script performs tasks before the Solaris software is installed on a system. You can use begin scripts only when using custom JumpStart to install the Solaris software.

Use a begin script to perform one of the following tasks:

- Create derived profiles
- Back up files before upgrading
Important Information About Begin Scripts

- Do not specify something in the script that would prevent the mounting of file systems onto /a during an initial or upgrade installation. If the JumpStart program cannot mount the file systems onto /a, an error occurs and installation fails.
- During the installation, output from the begin script is deposited in /tmp/begin.log. After the installation is completed, the log file is redirected to /var/sadm/system/logs/begin.log.
- Ensure that root owns the begin script and that the permissions are set to 644.
- You can use custom JumpStart Environment variables in your begin scripts. For a list of environment variables, see “Custom JumpStart Environment Variables” on page 308.
- 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}
for pkg in "package SUNWman delete" "package SUNWolman delete" "package SUNWxwman delete" done
```

EXAMPLE 24–1 Begin Script That Creates a Derived Profile  (Continued)

In the example, the begin script must use the SI_PROFILE environment variable for the name of the derived profile, which is set to /tmp/install.input by default.

Note – If a begin script is used to create a derived profile, ensure the script does not have any errors. A derived profile is not verified by the check script because derived profiles are not created until the execution of the begin script.

Creating Finish Scripts

A finish script is a user-defined Bourne shell script that you specify in the rules file. A finish script performs tasks after the Solaris software is installed on a system, but before the system reboots. You can use finish scripts only when using custom JumpStart to install Solaris.

Tasks that you can perform with a finish script include the following:

- Add files
- Add individual packages or patches in addition to the ones that are installed in a particular software group
- Customize the root environment
- Set the system’s root password
- Install additional software

Important Information About Finish Scripts

- The Solaris suninstall program mounts the system’s file systems on /a. The file systems remain mounted on /a until the system reboots. You can use the finish script to add, change, or remove files from the newly installed file system hierarchy by modifying the file systems that are respective to /a.
- During the installation, output from the finish script is deposited in /tmp/finish.log. After the installation is completed, the log file is redirected to /var/sadm/system/logs/finish.log.
- Ensure that root owns the finish script and that the permissions are set to 644.
- You can use custom JumpStart Environment variables in your finish scripts. For a list of environment variables, see “Custom JumpStart Environment Variables” on page 308.
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:

   ```bash
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:

   ```bash
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**

```bash
#!/bin/sh

BASE=/a
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin
```
EXAMPLE 24-2 Adding Packages With a Finish Script  

```sh
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
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:

```sh
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

```bash
#!/bin/sh
#
# Customize root’s environment
#
echo "***adding customizations in /.cshrc"
test -f a/.cshrc || {
cat >> a/.cshrc <<EOF
set history=100 savehist=200 filec ignoreeof prompt="\$user@'uname -n' > 
alias cp cp -i
alias mv mv -i
alias rm rm -i
alias ls ls -FC
alias h history
alias c clear
unset autologout
EOF
}
```

Setting a System’s Root Password With a Finish Script

After the Solaris software is installed on a system, the system reboots. Before the boot process is completed, the system prompts for the root password. Until someone types a password, the system cannot finish booting.

A finish script that is named `set_root_pw` is saved in the `auto_install_sample` directory. The finish script shows how to set the root password automatically, without prompting. `set_root_pw` is shown in Example 24-4.

EXAMPLE 24-4 Setting the System’s Root Password With a Finish Script

```bash
#!/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=dKO51BkSF421w
# create a temporary input file
cp /a/etc/shadow /a/etc/shadow.orig
mv /a/etc/shadow /a/etc/shadow.orig
nawk -F: '{
```
EXAMPLE 24–4 Setting the System’s Root Password With a Finish Script  (Continued)

```bash
if ( $1 == "root" )
    printf "%s:%s:%s:%s:%s:%s:%s:%s:%s\n", $1, passwd, $3, $4, $5, $6, $7, $8, $9
else
    printf "%s:%s:%s:%s:%s:%s:%s:%s:%s\n", $1, $2, $3, $4, $5, $6, $7, $8, $9
}' passwd="$PASSWD" /a/etc/shadow.orig > /a/etc/shadow
#remove the temporary file
del -f /a/etc/shadow.orig
# set the flag so sysidroot won’t prompt for the root password
del -e ‘s/0 # root/1 # root/’ ${SI_SYS_STATE} > /tmp/state.$$5
mv /tmp/state.$$ ${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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-nodisplay</td>
<td>Runs the installer without a graphic user interface. Use the default product installation unless the installation was modified by the -locales option.</td>
</tr>
<tr>
<td>-noconsole</td>
<td>Runs the installation without any interactive text console device. Useful when paired with -nodisplay for UNIX script use.</td>
</tr>
</tbody>
</table>

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.

   ```bash
   # 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**

```bash
# cd /jumpstart
# tar -cvf config.tar *
a profile 1K
a rules 1K
a rules.ok 1K
a sysidcfg 1K
```

Creating Disk Configuration Files

This section describes how to create single-disk and multiple-disk configuration files. Disk configuration files enable you to use `pfinstall(1M)` from a single system to test profiles against different disk configurations.

▼ SPARC: To Create a Disk Configuration File

1. Locate a SPARC system with a disk you want to test.

2. Become superuser.

3. Create a single-disk configuration file by redirecting the output of the `prtvtoc(1M)` command to a file.

   ```bash
   # prtvtoc /dev/rdsk/device_name >disk_config_file
   ```
The device name of the system’s disk. `device_name` must be in the form `cwtxds2` or `cxds2`.

disk_config_file

The name of the disk configuration file.

4. Determine if you are testing the installation of Solaris software on multiple disks.
   - If no, stop. You are finished.
   - If yes, concatenate the single-disk configuration files and save the output in a new file.
     ```
     # cat disk_file1 disk_file2 >multi_disk_config
     ```
     The new file becomes the multiple-disk configuration file, as in the following example:
     ```
     # cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
     ```

5. Determine if the target numbers in the disk device names are unique in the multiple-disk configuration file that you created in the previous step.
   - If yes, stop. You are finished.
   - If no, open the file with a text editor and make the target numbers unique in the disk device names.
     For example, assume that the file contains the same target number, t0, for different disk device names, as shown here:
     ```
     * /dev/rdsk/c0t0d0s2 partition map
     ...
     * /dev/rdsk/c0t0d0s2 partition map
     ```
     Change the second target number to t2, as shown here:
     ```
     * /dev/rdsk/c0t0d0s2 partition map
     ...
     * /dev/rdsk/c0t2d0s2 partition map
     ```

SPARC: Disk Configuration File Example

The following example shows how to create a single-disk configuration file, `104_test`, on a SPARC system with a 104-Mbyte disk.

**EXAMPLE 24–6 SPARC: Creating a Disk Configuration File**

You redirect the output of the `prtvtoc` command to a single-disk configuration file that is named `104_test`:
```
# prtvtoc /dev/rdsk/c0t3d0s2 >104_test
```

The contents of the `104_test` file resemble the following:
EXAMPLE 24-6 SPARC: Creating a Disk Configuration File  (Continued)

*/dev/rdsk/c0t3d0s2 partition map
*
* Dimensions:
*  512 bytes/sector
*  72 sectors/track
*  14 tracks/cylinder
*  1008 sectors/cylinder
*  2038 cylinders*  2036 accessible cylinders
* Flags:
*  1: unmountable
* 10: read-only
*
<table>
<thead>
<tr>
<th>Partition</th>
<th>Tag</th>
<th>Flags</th>
<th>Sector Count</th>
<th>Sector Map Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>00</td>
<td>164304</td>
<td>164303 /</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>00</td>
<td>2052288</td>
<td>2052287</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>00</td>
<td>164304</td>
<td>823536 987839 /disk2/b298</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>00</td>
<td>987840</td>
<td>614880 1602719 /install/298/sparc/work</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>00</td>
<td>1602720</td>
<td>449568 2052287 /space</td>
</tr>
</tbody>
</table>

You have created disk configuration files for a SPARC based system. “Testing a Profile” on page 229 contains information about using disk configuration files to test profiles.

▼ x86: To Create a Disk Configuration File

1. Locate an x86 based system that contains a disk that you are testing.

2. Become superuser.

3. Create part of the single-disk configuration file by saving the output of the fdisk(1M) command in a file.

```bash
# fdisk -R -W disk_config_file -h /dev/rdsk/device_name
```

- `disk_config_file` The name of a disk configuration file.
- `/dev/rdsk/device_name` The device name of the fdisk layout of the entire disk. `device_name` must be in the form `cxdyp0` or `cxdyp0`.

4. Append the output of the prtvtoc(1M) command to the disk configuration file:

```bash
# prtvtoc /dev/rdsk/device_name >>disk_config
```
5. Determine if you are testing the installation of Solaris software on multiple disks.
   - If no, stop. You are finished.
   - If yes, concatenate the single-disk configuration files and save the output in a new file:
     ```
     # cat disk_file1 disk_file2 >multi_disk_config
     ```
     The new file becomes the multiple-disk configuration file, as in the following example:
     ```
     # cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
     ```

6. Determine if the target numbers in the disk device names are unique in the multiple-disk configuration file that you created in the previous step.
   - If yes, stop. You are finished.
   - If no, open the file with a text editor and make the target numbers unique.
     For example, if the file contains the same target number, t0, for different disk device names as shown here:
     ```
     * /dev/rdsk/c0t0d0s2 partition map
     ...
     * /dev/rdsk/c0t0d0s2 partition map
     ```
     Change the second target number to t2, as shown here:
     ```
     * /dev/rdsk/c0t0d0s2 partition map
     ...
     * /dev/rdsk/c0t2d0s2 partition map
     ```

x86: Disk Configuration File Example

The following example shows how to create a single-disk configuration file, 500_test, on an x86 system that contains a 500-Mbyte disk.

**EXAMPLE 24-7 x86: Creating a Disk Configuration File**

First, you save the output of the fdisk command to a file that is named 500_test:
```
# fdisk -R -W 500_test -h /dev/rdsk/c0t0d0p0
```

The 500_test file looks like the following:
```
* /dev/rdsk/c0t0d0p0 default fdisk table
* Dimensions:
```
EXAMPLE 24–7 x86: Creating a Disk Configuration File  (Continued)

* 512 bytes/sector
* 94 sectors/track
* 15 tracks/cylinder
* 1455 cylinders
*
* HBA Dimensions:
* 512 bytes/sector
* 94 sectors/track
* 15 tracks/cylinder
* 1455 cylinders
*
* systid:
* 1: DOSOS12
* 2: PCIXOS
* 4: DOSOS16
* 5: EXTDS
* 6: DOSBIG
* 86: DOSDATA
* 98: OTHEROS
* 99: UNIXOS
* 130: SUNIXOS
*
* Id Act Bhead Bsect Bcyl Ehead Esect Ecyl Rsect Numsect
130 128 44 3 0 46 30 1001 1410 2050140

Second, you append the output of the prtvtoc command to the 500_test file:

```bash
# prtvtoc /dev/rdsk/c0t0d0s2 >>500_test
```

The 500_test file is now a complete disk configuration file:

```plaintext
/dev/rdsk/c0t0d0p0 default fdisk table
Dimensions:
* 512 bytes/sector
* 94 sectors/track
* 15 tracks/cylinder
* 1455 cylinders
*
* HBA Dimensions:
* 512 bytes/sector
* 94 sectors/track
* 15 tracks/cylinder
* 1455 cylinders
*
* systid:
* 1: DOSOS12
* 2: PCIXOS
* 4: DOSOS16
* 5: EXTDS
* 6: DOSBIG
* 86: DOSDATA
* 98: OTHEROS
* 99: UNIXOS
```
EXAMPLE 24–7 x86: Creating a Disk Configuration File  (Continued)

* 130: SUNIXOS
* 130 128 44 3 0 46 30 1001 1410 2050140
* /dev/rdsk/c0t0d0s2 partition map
* Dimensions:
  * 512 bytes/sector
  * 94 sectors/track
  * 15 tracks/cylinder
  * 1110 sectors/cylinder
  * 1454 cylinders
  * 1452 accessible cylinders
* Flags:
  * 1: unmountable
  * 10: read-only
* 1: unmountable
* 10: read-only
* 1: unmountable
* 10: read-only

<table>
<thead>
<tr>
<th>Partition</th>
<th>Tag</th>
<th>Flags</th>
<th>First Sector</th>
<th>Count</th>
<th>Last Sector</th>
<th>Mount Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>01</td>
<td>1410</td>
<td>2045910</td>
<td>2047319</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>00</td>
<td>4230</td>
<td>2043090</td>
<td>2047319</td>
<td>/space</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>01</td>
<td>0</td>
<td>1410</td>
<td>1409</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>01</td>
<td>1410</td>
<td>2820</td>
<td>422987</td>
<td></td>
</tr>
</tbody>
</table>

You have created disk configuration files for an x86 based system. “Testing a Profile” on page 229 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 249
- “Creating a custom_probes File” on page 250
- “Validating the custom_probes File” on page 253

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 311
Creating a custom_probes File

If the rule and probe keywords that are described in “Rule Keywords and Values” on page 277 and “Probe Keywords and Values” on page 311 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

```bash
#!/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**

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>check searches for a custom_probes file.</td>
</tr>
<tr>
<td>2</td>
<td>If the file exists, check creates the custom_probes.ok file from the custom_probes file, removes all comments and blank lines, and retains all Bourne shell commands, variables, and algorithms. Then, check adds the following comment line at the end: # version=2 checksum=num</td>
</tr>
</tbody>
</table>

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.
-x 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:
     ```bash
     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 on a SPARC based or an x86 based system. You need to follow these procedures on the system on which you intend to install the Solaris 9 software.

- “SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 258
- “x86: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 262

SPARC: Performing a Custom JumpStart Installation

SPARC: Task Map: Setting Up a System for a Custom JumpStart Installation

During a custom JumpStart installation, the JumpStart program attempts to match the system that is being installed to the rules in the rules.ok file. The JumpStart program reads the rules from the first rule through the last. A match occurs when the system that is being installed matches all the system attributes that are defined in the rule. As soon as a system matches a rule, the JumpStart program stops reading the rules.ok file and begins to install the system, based on the matched rule’s profile.
### TABLE 26–1 Task Map: Setting Up a System for a Custom JumpStart Installation

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check if the system is supported</td>
<td>Check the hardware documentation for system support in the Solaris 9 environment.</td>
<td>Solaris 9 Sun Hardware Platform Guide</td>
</tr>
<tr>
<td>Check if the system has enough disk space for the Solaris 9 software</td>
<td>Verify that you have planned enough space to install the Solaris software on your system.</td>
<td>Chapter 5</td>
</tr>
<tr>
<td>(Optional) Preconfigure system configuration information</td>
<td>You can use the sysidcfg file or the name service to preconfigure installation information for a system. If you preconfigure system information, the installation program does not prompt you to supply the information during the installation.</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>Prepare the system for custom JumpStart installation</td>
<td>Create and validate a rules file and profile files.</td>
<td>Chapter 23</td>
</tr>
<tr>
<td>(Optional) Prepare optional custom JumpStart features</td>
<td>If you are using begin scripts, finish scripts, or other optional features, prepare the scripts or files.</td>
<td>Chapter 24 and Chapter 25</td>
</tr>
<tr>
<td>(Optional) Set up the system to install over the network</td>
<td>To install a system from a remote Solaris 9 DVD or Solaris 9 Software SPARC Platform Edition CD image, you need to set up the system to boot and install from an install server or a boot server.</td>
<td>Chapter 12</td>
</tr>
<tr>
<td>(Optional) Prepare for a Solaris Flash archive installation</td>
<td>Before installing a Solaris Flash archive, you need to follow specific steps.</td>
<td>“To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 257</td>
</tr>
<tr>
<td>Install or upgrade</td>
<td>Boot the system to initiate the installation or upgrade.</td>
<td>“SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 258</td>
</tr>
</tbody>
</table>
To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation

You can install a full archive or a differential archive using custom JumpStart. For an overview of a full or differential archive, see Chapter 17.

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.
   For examples of Solaris Flash archive profiles, see “Profile Examples” on page 226.

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

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Initial installation</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>archive_location</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>fdisk (x86 only)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>filesystem – You cannot set the filesystem keyword to the value auto</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>forced_deployment</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>install_type — required</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>local_customization</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>no_content_check</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>no_master_check</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>root_device</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

a. Set the value of the keyword install_type to one of the following types.
   - For a full archive installation, set the value to flash_install
   - For a differential archive installation, set the value to flash_update

b. Add the path to the Solaris Flash archive by using the archive_location keyword.
   Refer to “archive_location Keyword” on page 283 for details about the archive_location keyword.

c. Specify the file system configuration.
The Solaris Flash archive extraction process does not support auto-layout of partitions.

d. (Optional) If you want to install an additional Solaris Flash archive on the clone system, add one archive_location line for each archive that you want to install.

3. On the install server, add the clients that you are installing with the Solaris Flash archive.
   For detailed instructions, refer to “Adding Systems to Be Installed From the Network With a DVD Image” on page 91 or “Adding Systems to Be Installed From the Network With a CD Image” on page 119.

4. Perform the custom JumpStart installation on the clone systems.
   For detailed instructions, refer to “SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 258

▼ SPARC: To Perform an Installation or Upgrade With the Custom JumpStart Program

1. If the system is part of a network, ensure that an Ethernet connector or similar network adapter is attached to your system.

2. If you are installing a system that is connected through a tip(1) line, ensure that your window display is at least 80 columns wide and 24 rows long.
   To determine the current dimensions of your tip window, use the stty(1) command.

3. If you are using the system’s DVD-ROM or CD-ROM drive to install the Solaris 9 software, insert the Solaris 9 SPARC Platform Edition DVD or the Solaris 9 Software 1 of 2 SPARC Platform Edition CD in the drive.

4. If you are using a profile diskette, insert the profile diskette in the system’s diskette drive.

5. Boot the system.
   □ If the system is new, out-of-the-box, turn on the system.
   □ If you want to install or upgrade an existing system, shut down the system. At the ok prompt, type the following command:

```
ok boot cdrom:net - install [url:ask] [dhcp] [nowin]
```
cdrom  Specifies to boot from a CD or a DVD.

For a system with an older EEPROM, replace cdrom with sd(0,6,2) to boot from the system's CD-ROM or DVD-ROM drive.

net  Specifies to boot from an install server on the network.

url  Specifies the location of the custom JumpStart files. You can specify a URL for files that are located in the following places:

- Local hard disk
  
  file://jumpstart_dir_path/compressed_config_file

- NFS server
  
  nfs://server_name:IP_address/jumpstart_dir_path/compressed_config_file

- HTTP server
  
  http://server_name:IP_address/jumpstart_dir_path/compressed_config_file&proxy_info

If you placed a sysidcfg file in the compressed configuration file, you must specify the IP address of the server that contains the file, as in the following example:

http://131.141.2.32/jumpstart/config.tar

If you saved the compressed configuration file on an HTTP server that is behind a firewall, you must use a proxy specifier during boot. You do not need to specify an IP address for the server that contains the file. You must specify an IP address for the proxy server, as in the following example:

http://www.shadow.com/jumpstart/config.tar&proxy=131.141.6.151

ask  Specifies that the installation program prompt you to type the location of the compressed configuration file after the system boots and connects to the network.

If you bypass the prompt by pressing Return, the installation program interactively configures the network parameters. The installation program then prompts you for the location of the compressed configuration file. If you bypass the prompt by pressing Return, the Solaris suninstall program begins.

dhcp  Specifies to use a DHCP server to obtain network installation information that is needed to boot the system.

If you do not specify to use a DHCP server, the system uses the /etc/bootsparams file or the name service bootparams database.

nowin  Specifies not to begin the X program. You do not need to use the X program to perform a custom JumpStart installation, so you can reduce the installation time by using the nowin option.
Note – SPARC: The system checks hardware and system components and your SPARC system boots. Booting lasts several minutes.

6. If you did not preconfigure system information in the \texttt{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:
   - \texttt{/var/sadm/system/logs}
   - \texttt{/var/sadm/install/logs}

\textbf{x86: Performing a Custom JumpStart Installation}

\textbf{x86: Task Map: Setting Up a System for a Custom JumpStart Installation}

During a custom JumpStart installation, the JumpStart program attempts to match the system that is being installed to the rules in the \texttt{rules.ok} file. The JumpStart program reads the rules from the first rule through the last rule. A match occurs when the system that is being installed matches all of the system attributes that are defined in the rule. As soon as a system matches a rule, the JumpStart program stops reading the \texttt{rules.ok} file and begins to install the system, based on the matched rule’s profile.
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine if you need to preserve an existing operating system and user data</td>
<td>If the existing operating system on the system uses the entire disk, you must preserve the existing operating system so it can co-exist with the Solaris 9 software. This decision determines how to specify the <code>fdisk(1M)</code> keyword in the system's profile.</td>
<td>“x86: fdisk Profile Keyword” on page 294</td>
</tr>
<tr>
<td>Check if the system is supported</td>
<td>Check the hardware documentation for system support in the Solaris 9 environment.</td>
<td>Hardware manufacturer’s documentation</td>
</tr>
<tr>
<td>Check if the system has enough disk space for the Solaris 9 software</td>
<td>Verify that you have planned enough space to install the Solaris software on your system.</td>
<td>Chapter 5</td>
</tr>
<tr>
<td>(Optional) Preconfigure system configuration information</td>
<td>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.</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>Prepare the system for custom JumpStart installation</td>
<td>Create and validate a <code>rules</code> file and profile files.</td>
<td>Chapter 23</td>
</tr>
<tr>
<td>(Optional) Prepare optional custom JumpStart features</td>
<td>If you are using <code>begin</code> scripts, <code>finish</code> scripts, or other optional features, prepare the scripts or files.</td>
<td>Chapter 24 and Chapter 25</td>
</tr>
<tr>
<td>(Optional) Set up the system to install over the network</td>
<td>To install a system from a remote Solaris 9 x86 Platform Edition DVD or Solaris 9 Software x86 Platform Edition CD image, you need to set up the system to boot and install from an install server or a boot server.</td>
<td>Chapter 12</td>
</tr>
</tbody>
</table>
TABLE 26–2 x86: Task Map: Setting Up a System for a Custom JumpStart Installation
(Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install or upgrade</td>
<td>Boot the system to initiate the installation or upgrade.</td>
<td>“x86: To Perform an Installation or Upgrade With the Custom JumpStart Program” on page 262</td>
</tr>
</tbody>
</table>

▼ x86: To Perform an Installation or Upgrade With the Custom JumpStart Program

1. If the system is part of a network, ensure that an Ethernet connector or similar network adapter is attached to your system.

2. If you want to install a system that is connected through a `tip(1)` line, ensure that your window display is at least 80 columns wide and 24 rows long.
   To determine the current dimensions of your `tip` window, use the `stty(1)` command.

3. If you are using a profile diskette, insert the profile diskette into the system’s diskette drive.

   **Note** – The profile diskette contains a copy of the Solaris 9 Device Configuration Assistant in addition to profile information. If you are using PXE network boot to boot the system over the network, you must configure your system so that the system boots from the network and not from the diskette.

4. If you are using the system’s DVD-ROM or CD-ROM drive to install the Solaris 9 software, insert the Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 1 of 2 x86 Platform Edition CD in the drive.

5. Decide how to boot the system.
   - If you boot from the Solaris 9 DVD or the Solaris 9 Installation CD, insert the disc. Your system’s BIOS must support booting from a DVD or CD.
   - If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the system’s BIOS setup tool or the network adapter’s configuration setup tool.
   - If you boot from a Diskette – Use the profile diskette that you inserted into the drive in Step 3 or insert the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette into the system’s diskette drive.
**x86 only** – You can copy the Device Configuration Assistant software to a diskette by the following methods:

- Copy the software to a diskette from the Solaris 9 x86 Platform Edition DVD or Solaris 9 Software 2 of 2 x86 Platform Edition CD by using the procedure in Appendix E.

6. **If the system is off,** turn the system on. **If the system is on,** reboot the system.
   The Device Configuration Assistant identifies the system’s devices.

7. **On the Boot Solaris screen,** select the device from which to boot the system. Select DVD, CD, Net, or Disk.

8. **At the prompt,** perform one of the following:
   Select the type of installation you want to perform:
   
   1. Solaris Interactive
   2. Custom JumpStart

   Enter the number of your choice followed by the <ENTER> key.

   If you enter anything else, or if you wait for 30 seconds, an interactive installation will be started.

   To select the custom JumpStart method, perform one of the following actions:

   **Note** – If you do not type 2 or type a boot command within 30 seconds, the Solaris suninstall program begins. You can stop the timer by typing any key on the command line.

   - Type 2 and press Enter.
   - To specify the location of the custom JumpStart configuration files, type the following command:

     ```
b install [url:ask] [dhcp] [nowin]
```
url

Specifies the location of the custom JumpStart files. You can specify a URL for files that are located in the following places:

- Local hard disk
  file://jumpstart_dir_path/compressed_config_file
- NFS server
  nfs://server_name:IP_address/jumpstart_dir_path/compressed_config_file
- HTTP server
  http://server_name:IP_address/jumpstart_dir_path/
  compressed_config_file?proxy_info

If you placed a sysidcfg file in the compressed configuration file, you must specify the IP address of the server that contains the file, as in the following example:

http://131.141.2.32/jumpstart/config.tar

If you saved the compressed configuration file on an HTTP server that is behind a firewall, you must use a proxy specifier during boot. You do not need to specify an IP address for the server that contains the file. You must specify an IP address for the proxy server, as in the following example:

http://www.shadow.com/jumpstart/
  config.tar?proxy=131.141.6.151

ask

Specifies that the installation program prompt you to type the location of the compressed configuration file after the system boots and connects to the network.

If you bypass the prompt by pressing Return, the installation program interactively configures the network parameters. The installation program then prompts you for the location of the compressed configuration file. If you bypass the prompt by pressing Return, the Solaris suninstall program begins.

dhcp

Specifies to use a DHCP server to obtain network installation information that is needed to boot the system.

If you do not specify to use a DHCP server, the system uses the /etc/bootparams file or the name service bootparams database.

nowin

Specifies not to begin the X program. You do not need to use the X program to perform a custom JumpStart installation, so you can reduce the installation time by using the nowin option.

9. If you did not preconfigure system information in the sysidcfg file, when prompted, answer the questions about system configuration.

10. Follow the instructions on the screen to install the software.

   When the JumpStart program finishes installing the Solaris software, the system reboots automatically.
After the installation is finished, installation logs are saved in a file. You can find the installation logs in the following directories:

- /var/sadm/system/logs
- /var/sadm/install/logs
This chapter provides an example of setting up and installing Solaris software on both SPARC based and x86 based systems by using a custom JumpStart installation.

- “Sample Site Setup” on page 267
- “Create an Install Server” on page 269
- “x86: Create a Boot Server for Marketing Systems” on page 270
- “Create a JumpStart Directory” on page 271
- “Share the JumpStart Directory” on page 271
- “SPARC: Create the Engineering Group’s Profile” on page 271
- “x86: Create the Marketing Group’s Profile” on page 272
- “Update the rules File” on page 272
- “Validate the rules File” on page 273
- “SPARC: Set Up Engineering Systems to Install From the Network” on page 273
- “x86: Set Up Marketing Systems to Install From the Network” on page 274
- “SPARC: Boot the Engineering Systems and Install Solaris 9 Software” on page 275
- “x86: Boot the Marketing Systems and Install Solaris 9 Software” on page 275

Sample Site Setup

Figure 27–1 shows the site setup for this example.
At this sample site, the conditions are as follows:

- **SPARC**: The engineering group is located on its own subnet. This group uses SPARCstation™ systems for software development.
- **x86**: The marketing group is located on its own subnet. This group uses x86 based systems for running word processors, spreadsheets, and other office productivity tools.
- The site uses NIS. The Ethernet addresses, IP addresses, and host names of the systems are preconfigured in the NIS maps. The subnet mask, date and time, and geographic region for the site are also preconfigured in the NIS maps.

**Note** – The peripheral devices for the marketing systems are preconfigured in the `sysidcfg` file.

- Both the engineering and marketing systems are to be installed with Solaris 9 software from the network.
Create an Install Server

Because the groups need to install Solaris 9 software from the network, you make server-1 an install server for both groups. You use the `setup_install_server(1M)` command to copy the images to the server-1 local disk (in the `/export/install` directory). Copy the images from the Solaris 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` and `x86_9` directories.

**EXAMPLE 27-1 SPARC: Copying the Solaris 9 CDs**

Insert the Solaris 9 Software 1 of 2 **SPARC Platform Edition** CD in the CD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# mkdir -p /export/install/sparc_9
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9
```

Insert the Solaris 9 Software 2 of 2 **SPARC Platform Edition** CD in the CD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

Insert the Solaris 9 **SPARC Platform Edition** Languages CD in the CD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

**EXAMPLE 27-2 x86: Copying the Solaris 9 CDs**

Insert the Solaris 9 Software 1 of 2 **x86 Platform Edition** CD in the CD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# mkdir -p /export/install/x86_9
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/x86_9
```

Insert the Solaris 9 Software 2 of 2 **x86 Platform Edition** CD in the CD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/x86_9
```

Insert the Solaris 9 **x86 Platform Edition** Languages CD in the CD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/x86_9
```
EXAMPLE 27-3 SPARC: Copying the Solaris 9 DVD

Insert the Solaris 9 SPARC Platform Edition DVD in the DVD-ROM drive that is attached to server-1 and type the following commands:

server-1# mkdir -p /export/install/sparc_9
server-1# cd /DVD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9

EXAMPLE 27-4 x86: Copying the Solaris 9 DVD

Insert the Solaris 9 x86 Platform Edition DVD in the DVD-ROM drive that is attached to server-1 and type the following commands:

server-1# mkdir -p /export/install/x86_9
server-1# cd /DVD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/x86_9

x86: Create a Boot Server for Marketing Systems

Systems cannot boot from an install server on a different subnet, so you make server-2 a boot server on the marketing group’s subnet. You use the setup_install_server(1M) command to copy the boot software from the Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 1 of 2 x86 Platform Edition CD to the server-2 local disk (in the /export/boot directory).

If you insert the Solaris 9 Software 1 of 2 x86 Platform Edition CD in the CD-ROM drive that is attached to server-2, type the following command:

server-2# cd /CD_mount_point/Solaris_9/Tools
server-2# ./setup_install_server -b /export/boot

If you insert the Solaris 9 x86 Platform Edition DVD in the DVD-ROM drive that is attached to server-2, type the following command:

server-2# cd /DVD_mount_point/Solaris_9/Tools
server-2# ./setup_install_server -b /export/boot

In the setup_install_server command, -b specifies that setup_install_server is to copy the boot information to the directory that is named /export/boot.
Create a JumpStart Directory

Now that you have the install server and boot server set up, you create a JumpStart directory on server-1. You can use any system on the network. This directory holds files that are required for a custom JumpStart installation of Solaris software. You set up this directory by copying the sample directory from the Solaris 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

SPARC: Create the Engineering Group’s Profile

For the engineering systems, you create a file that is named eng_prof in the /jumpstart directory. The eng_prof file contains the following entries, which define the Solaris 9 software to be installed on systems in the engineering group:

install_type initial_install
system_type standalone
partitioning default
cluster SUNWCprog
filesys any 512 swap

1. Specifies that the installation is to be treated as an initial installation, as opposed to an upgrade.
2. Specifies that the engineering systems are standalone systems.
3. Specifies that the JumpStart software uses default disk partitioning for installing Solaris software on the engineering systems.
4. Specifies that the Developer System Support software group is to be installed.
5. Specifies that each system in the engineering group is to have 512 Mbytes of swap space.

**x86: Create the Marketing Group’s Profile**

For the marketing systems, you create a file that is named `marketing_prof` in the `/jumpstart` directory. The `marketing_prof` file contains the following entries, which define the Solaris 9 software to be installed on systems in the marketing group:

```
install_type initial_install
system_type standalone
partitioning default
cluster SUNWCuser
package SUNWaudio
```

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 subnet 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.

---

**SPARC: Set Up Engineering Systems to Install From the Network**

After setting up the /jumpstart directory and files, you use the add_install_client command on the install server, server-1, to set up the engineering systems to install the Solaris 9 software from the install server. server-1 is also the boot server for the engineering group’s subnet.

```
server-1# cd /export/install/sparc_9/Solaris_9/Tools
server-1# ./add_install_client -c server-1:/jumpstart host-eng1 sun4u
```
server-1# ./add_install_client -c server-1:/jumpstart host-eng2 sun4u

In the add_install_client command, the options that are used have the following meanings:

- \texttt{-c} Specifies the server (server-1) and path (/jumpstart) to the JumpStart directory.

- host-eng1 The name of a system in the engineering group.

- host-eng2 The name of another system in the engineering group.

- sun4u Specifies the platform group of the systems that use server-1 as an install server. The platform group is for Ultra 5 systems.

\section*{x86: Set Up Marketing Systems to Install From the Network}

Next, you use the add_install_client command on the boot server (server-2). This command sets up the marketing systems to boot from the boot server and install the Solaris 9 software from the install server (server-1):

\begin{verbatim}
server-2# cd /marketing/boot-dir/Solaris_9/Tools
server-2# ./add_install_client -s server-1:/export/install/x86_9 \
   -c server-1:/jumpstart host-mkt1 i86pc
server-2# ./add_install_client -s server-1:/export/install/x86_9 \
   -c server-1:/jumpstart host-mkt2 i86pc
server-2# ./add_install_client -d -s server-1:/export/install/x86_9 \
   -c server-1:/jumpstart SUNW.i86pc i86pc
\end{verbatim}

In the add_install_client command, the options that are used have the following meanings:

- \texttt{-d} Specifies that the client is to use DHCP to obtain the network install parameters. This option is required for clients to use PXE network boot to boot from the network. -d is optional for network boot clients that do not use PXE network boot.

- \texttt{-s} Specifies the install server (server-1) and the path to the Solaris 9 software (/export/install/x86_9).

- \texttt{-c} Specifies the server (server-1) and path (/jumpstart) to the JumpStart directory.
SPARC: Boot the Engineering Systems and Install Solaris 9 Software

After setting up the servers and files, you can boot the engineering systems by using the following `boot` command at the `ok` (PROM) prompt of each system:

```
ok boot net - install
```

The Solaris operating environment is automatically installed on the engineering group’s systems.

---

x86: Boot the Marketing Systems and Install Solaris 9 Software

You can boot the system from one of the following:

- Solaris 9 Software 1 of 2 x86 Platform Edition CD
- Solaris 9 x86 Platform Edition DVD
- The network by using PXE network boot
- The profile diskette
- The Solaris 9 Device Configuration Assistant x86 Platform Edition diskette

Solaris 9 is automatically installed on the marketing group’s 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 277
- “Profile Keywords and Values” on page 282
- “Custom JumpStart Environment Variables” on page 308
- “Probe Keywords and Values” on page 311

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 221.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>minus sign (-)</td>
<td>Anything. The any keyword always succeeds.</td>
</tr>
<tr>
<td>arch</td>
<td>processor_type</td>
<td>A system’s processor type.</td>
</tr>
<tr>
<td></td>
<td>Valid values for processor_type are the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ SPARC: sparc</td>
<td>The uname -p command reports the system’s processor type.</td>
</tr>
<tr>
<td></td>
<td>■ x86: i386</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Value</td>
<td>Matches</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>disksize</code></td>
<td><code>actual_disk_name size_range</code></td>
<td>The name and size of a system’s disk in Mbytes.</td>
</tr>
<tr>
<td></td>
<td><em>actual_disk_name</em> – A disk name in the form cxtzydz, such as c0t3d0 or c0d0, or the special word <code>rootdisk</code>. If <code>rootdisk</code> is used, the disk to be matched is determined in the following order:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPARC: The disk that contains the preinstalled boot image, which is a new SPARC based system with factory JumpStart installed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The c0t3d0s0 disk, if the disk exists</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The first available disk that is searched in kernel probe order</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>size_range</em> – The size of the disk, which must be specified as a range of Mbytes (x-x).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>disksize c0t3d0 250-300</td>
</tr>
<tr>
<td></td>
<td>In the example, the JumpStart program attempts to match a system disk that is named c0t3d0. The disk can hold between 250 and 300 Mbytes of information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>disksize rootdisk 750-1000</td>
</tr>
<tr>
<td></td>
<td>In the example, the JumpStart program first attempts to match a system disk that contains a preinstalled boot image. Next, the JumpStart program attempts to match the c0t3d0s0 disk, if the disk exists. Finally, the JumpStart program attempts to match the first available disk that can hold between 750 Mbytes and 1 Gbyte of information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> – When calculating <em>size_range</em>, 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 <em>size_range</em> equal to 530–550.</td>
<td></td>
</tr>
<tr>
<td><code>domainname</code></td>
<td><code>actual_domain_name</code></td>
<td>A system’s domain name, which controls how a name service determines information.</td>
</tr>
<tr>
<td></td>
<td>If you have a system already installed, the <code>domainname</code> command reports the system’s domain name.</td>
<td></td>
</tr>
<tr>
<td><code>hostaddress</code></td>
<td><code>actual_IP_address</code></td>
<td>A system’s IP address.</td>
</tr>
<tr>
<td></td>
<td>If you have a system that is already installed, the <code>uname -n</code> command reports the system’s host name.</td>
<td></td>
</tr>
<tr>
<td><code>hostname</code></td>
<td><code>actual_host_name</code></td>
<td>A system’s host name.</td>
</tr>
</tbody>
</table>

**TABLE 28–1** Descriptions of Rule Keywords and Values (Continued)
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>installed</td>
<td>slice version</td>
<td>A disk that has a root (/) file system that corresponds to a particular version of Solaris software. Example: installed c0t3d0s1 Solaris_9 In the example, the JumpStart program attempts to match a system that has a Solaris 9 root (/) file system on c0t3d0s1.</td>
</tr>
<tr>
<td></td>
<td>slice</td>
<td>A disk slice name in the form cwtxyzab, such as c0t3d0s5, or the special words any or rootdisk. If any is used, the JumpStart program attempts to match all of the system’s disks in kernel probe order. If rootdisk is used, the disk to be matched is determined in the following order: SPARC: The disk that contains the preinstalled boot image, which is a new SPARC based system with factory JumpStart installed The c0t3d0s0 disk, if the disk exists The first available disk that is searched in kernel probe order.</td>
</tr>
<tr>
<td></td>
<td>version</td>
<td>A version name or the special words any or upgrade. If any is used, any Solaris or SunOS release is matched. If upgrade is used, any Solaris 2.1 or compatible release that can be upgraded is matched. If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is SystemV.</td>
</tr>
<tr>
<td>karch</td>
<td>actual_platform_group</td>
<td>A system’s platform group. If you have a system that is already installed, the arch -k command or the uname -m command reports the system’s platform group.</td>
</tr>
<tr>
<td>memsize</td>
<td>physical_mem</td>
<td>A system’s physical memory size in Mbytes. Example: memsize 64-128 The example tries to match a system with a physical memory size between 64 and 128 Mbytes. If you have a system that is already installed, the output of the prtconf command, line 2, reports the system’s physical memory size.</td>
</tr>
<tr>
<td>Keyword</td>
<td>Value</td>
<td>Matches</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>model</td>
<td>actual_platform_name</td>
<td>A system’s platform name. See the Solaris 9 Sun Hardware Platform Guide for a list of valid platform names.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong> — If the <code>actual_platform_name</code> contains spaces, you must replace spaces with underscores ( _ ).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example: SUNW,Sun_4_50</td>
</tr>
<tr>
<td>network</td>
<td>network_num</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example: network 193.144.2.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The example tries to match a system with a 193.144.2.8 IP address, if the subnet mask is 255.255.255.0.</td>
</tr>
<tr>
<td>osname</td>
<td>Solaris_x</td>
<td>A version of Solaris software that is already installed on a system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example: osname Solaris_9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In the example, the JumpStart program attempts to match a system with the Solaris 9 operating environment already installed.</td>
</tr>
</tbody>
</table>
### TABLE 28-1 Descriptions of Rule Keywords and Values (Continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>probe</td>
<td>探针_keyword</td>
<td>一个有效的探针关键词或有效的自定义探针关键词。</td>
</tr>
<tr>
<td></td>
<td></td>
<td>例子：探针磁盘</td>
</tr>
<tr>
<td></td>
<td></td>
<td>该例子返回系统磁盘的大小，单位为M字节，并按内核探针顺序排列，例如，<code>c0t3d0s1</code>, <code>c0t4d0s0</code>，在一个SPARC系统上。JumpStart程序设置<code>SI_DISKLIST</code>, <code>SI_DISKSIZEs</code>, <code>SI_NUMDISKS</code>, and <code>SI_TOTALDISK</code>环境变量。</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>注意</strong>—探针关键词是唯一的，因为该关键词不试图匹配一个属性并运行一个脚本。探针关键词返回一个值。因此，你不能指定开始脚本、配置文件以及完成脚本与探针规则关键词。探针关键词在第25章中描述。</td>
</tr>
<tr>
<td>totaldisk</td>
<td>size_range</td>
<td>磁盘空间的总和。磁盘空间包括所有操作磁盘，这些磁盘附属于一个系统。</td>
</tr>
<tr>
<td></td>
<td></td>
<td>例子：总磁盘 300-500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>在例子中，JumpStart程序试图匹配一个有总磁盘空间在300和500M字节之间的系统。</td>
</tr>
</tbody>
</table>
|         |           |**注意**—当计算size_range时，记住一个M字节等于1,048,576字节。一个磁盘可能被广告为一个“535-M字节”磁盘，但该磁盘可能只有510亿字节的磁盘空间。JumpStart程序将“535-M字节”磁盘视为510-M字节磁盘，因为535,000,000 / 1,048,576 = 510。一个“535-M字节”磁盘不匹配一个size_range等于530-550。
Profile Keywords and Values

This section describes the profile keywords and values that you can use in a profile. For detailed instructions to create a profile, see “Creating a Profile” on page 225.

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>
<thead>
<tr>
<th>Table 28–2 Overview of Profile Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profile Keywords</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>archive_location</strong> (Installing Solaris Flash archives)</td>
</tr>
<tr>
<td><strong>backup_media</strong></td>
</tr>
<tr>
<td><strong>boot_device</strong></td>
</tr>
<tr>
<td><strong>client_arch</strong></td>
</tr>
<tr>
<td><strong>client_root</strong></td>
</tr>
<tr>
<td><strong>client_swap</strong></td>
</tr>
<tr>
<td><strong>cluster (adding software groups)</strong></td>
</tr>
<tr>
<td><strong>cluster (adding or deleting clusters)</strong></td>
</tr>
<tr>
<td><strong>dontuse</strong></td>
</tr>
<tr>
<td><strong>fdisk</strong> (x86 only)</td>
</tr>
<tr>
<td><strong>filesystems (mounting remote file systems)</strong></td>
</tr>
<tr>
<td><strong>filesystems (creating local file systems)</strong></td>
</tr>
</tbody>
</table>

Solaris 9 Installation Guide • April 2003
TABLE 28–2 Overview of Profile Keywords  (Continued)

<table>
<thead>
<tr>
<th>Profile Keywords</th>
<th>Installation Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standalone System (Non-Networked)</td>
</tr>
<tr>
<td>forced_deployment (installing Solaris Flash differential archives)</td>
<td>✓</td>
</tr>
<tr>
<td>geo</td>
<td>✓</td>
</tr>
<tr>
<td>install_type</td>
<td>✓</td>
</tr>
<tr>
<td>isa_bits</td>
<td>✓</td>
</tr>
<tr>
<td>layout_constraint</td>
<td>✓</td>
</tr>
<tr>
<td>local_customization (installing Solaris Flash archives)</td>
<td>✓</td>
</tr>
<tr>
<td>locale</td>
<td>✓</td>
</tr>
<tr>
<td>no_master_check (installing Solaris Flash differential archives)</td>
<td>✓</td>
</tr>
<tr>
<td>no_content_check (installing Solaris Flash differential archives)</td>
<td>✓</td>
</tr>
<tr>
<td>num_clients</td>
<td>✓</td>
</tr>
<tr>
<td>package</td>
<td>✓</td>
</tr>
<tr>
<td>partitioning</td>
<td>✓</td>
</tr>
<tr>
<td>root_device</td>
<td>✓</td>
</tr>
<tr>
<td>system_type</td>
<td>✓</td>
</tr>
<tr>
<td>usedisk</td>
<td>✓</td>
</tr>
</tbody>
</table>

Profile Keyword Descriptions and Examples

archive_location Keyword

archive_location retrieval_type location

The values of retrieval_type and location depend on where the Solaris Flash archive is stored. The following sections contain the values you can use for retrieval_type and location and examples of how to use the archive_location keyword.
If the archive is stored on an NFS server, use the following syntax for the `archive_location` keyword.

```
archive_location nfs server_name:/path/ filename retry n
```

- **server_name**: The name of the server where you stored the archive.
- **path**: The location of the archive to be retrieved from the specified server. If the path contains $HOST, the Solaris Flash installation utilities replace $HOST with the name of the clone system that you are installing.
- **filename**: The name of the Solaris Flash archive file.
- **retry n**: An optional keyword. n is the maximum number of times the Solaris Flash utilities attempt to mount the archive.

**Examples:**

```
archive_location nfs golden:/archives/usrarchive
archive_location nfs://golden/archives/usrarchive
```

**HTTP 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
```

- **server_name**: The name of the server where you stored the archive. `server_name` can be a port number or the name of a TCP service that has a port number that is determined at runtime.
- **port**: An optional port. If you do not specify a port, the Solaris Flash installation utilities use the default HTTP port number, 80.
path

The location of the archive to be retrieved from the specified server. If the path contains $HOST, the Solaris Flash installation utilities replace $HOST with the name of the clone system that you are installing.

filename

The name of the Solaris Flash archive file.

optional_keywords

The optional keywords that you can specify when you retrieve a Solaris Flash archive from an HTTP server.

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Value Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth basic user_name password</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>timeout min</td>
<td>The timeout keyword enables you to specify, in minutes, the maximum length of time that is allowed to pass without receipt of data from the HTTP server before the connection is closed, reopened, and resumed from the point where the timeout occurred. If you specify a timeout value of 0 (zero), the connection is not reopened because of inactivity.</td>
</tr>
<tr>
<td></td>
<td>If a time-out reconnection occurs, the Solaris Flash installation utilities attempt to resume the installation at the last known position in the archive. If the Solaris Flash installation utilities cannot resume the installation at the last known position, the retrieval restarts from the beginning of the archive and the data that was retrieved prior to the timeout is discarded.</td>
</tr>
<tr>
<td>proxy host:port</td>
<td>The proxy keyword allows you to specify a proxy host and proxy port. You can use a proxy host to retrieve a Solaris Flash archive from the other side of a firewall. You must supply a proxy port when you specify the proxy keyword.</td>
</tr>
</tbody>
</table>

Example:

archive_location http://silver/archives/usrarchive.flar timeout 5

Example of the auth basic user_name password keyword:

archive_location http://silver/archives/usrarchive.flar timeout 5 user1 secret

FTP Server

If the archive is stored on an FTP server, use the following syntax for the archive_location keyword.

archive_location ftp://user_name:password@server_name:port path/filename optional_keywords
user_name:password  The user name and password that you need to access the FTP server
in the profile file.

server_name  The name of the server where you stored the archive. server_name
can be a port number or the name of a TCP service that has a port
number that is determined at runtime.

port  A is an optional port. If you do not specify a port, the Solaris Flash
installation utilities use the default FTP port number, 21.

path  The location of the archive to be retrieved from the specified server.
If the path contains $HOST, the Solaris Flash installation utilities
replace $HOST with the name of the clone system that you are
installing.

filename  The name of the Solaris Flash archive file.

optional_keywords  The optional keywords that you can specify when you retrieve a
Solaris Flash archive from an FTP server.

### TABLE 28–4 Optional Keywords to Use With archive_location ftp

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Value Definitions</th>
</tr>
</thead>
</table>
| timeout min  | The **timeout** keyword enables you to specify, in minutes, the maximum length of time that is allowed to pass without receipt of data from the FTP server
before the connection is closed, reopened, and resumed from the point where
the timeout occurred. If you specify a **timeout** value of 0 (zero), the
connection is not reopened because of inactivity.

If a time-out reconnection occurs, the Solaris Flash installation utilities attempt
to resume the installation at the last known position in the archive. If the Solaris
Flash installation utilities cannot resume the installation at the last known
position, the retrieval restarts from the beginning of the archive and the data
that was retrieved prior to the timeout is discarded. |
| proxy host:port | The **proxy** keyword allows you to specify a proxy host and proxy port. You can use a proxy host to retrieve a Solaris Flash archive from the other side of a
firewall. You must supply a proxy port when you specify the **proxy** keyword. |

Example:

archive_location ftp://user1:secret@silver/archives/usrarchive.flar timeout 5

**Local Tape**

If the archive is stored on a tape, use the following syntax for the
archive_location keyword.

archive_location local_tape device position
device
The name of the tape drive where you stored the Solaris Flash archive. If the device name is a canonical path, the Solaris Flash installation utilities retrieve the archive from the path to the device node. If you supply a device name that is not a canonical path, the Solaris Flash installation utilities add /dev/rmt/ to the path.

position
Designates the place on the tape drive where you saved the archive. If you do not supply a position, the Solaris Flash installation utilities retrieve the archive from the current position on the tape drive. By specifying a position, you can place a begin script or a sysidcfg file on the tape drive before the archive.

Examples:
archive_location local_tape /dev/rmt/0n 5
archive_location local_tape 0n 5

Local Device
You can retrieve a Solaris Flash archive from a local device if you stored the Solaris Flash archive on a file system-oriented, random-access device, such as a diskette or a CD-ROM. Use the following syntax for the archive_location keyword.

Note – You can retrieve an archive from stream-oriented devices, such as tape, by using the syntax for local tape.

archive_location local_device device path/filename file_system_type

device
The name of the drive where you stored the Solaris Flash archive. If the device name is a canonical path, the device is mounted directly. If you supply a device name that is not a canonical path, the Solaris Flash installation utilities add /dev/dsk/ to the path.

path
The path to the Solaris Flash archive, relative to the root of the file system on the device you specified. If the path contains $HOST, the Solaris Flash installation utilities replace $HOST with the name of the clone system that you are installing.

filename
The name of the Solaris Flash archive file.

file_system_type
Specifies the type of file system on the device. If you do not supply a file system type, the Solaris Flash installation utilities attempt to mount a UFS file system. If the UFS mount fails, the Solaris Flash installation utilities attempt to mount an HSFS file system.

Examples:
To retrieve an archive from a local hard drive that is formatted as a UFS file system, use the following command:

```bash
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:

```bash
archive_location local_device c0t0d0s0 /archives/usrarchive
```

**Local File**

You can retrieve an archive that you stored in the miniroot from which you booted the clone system as a local file. When you perform a custom JumpStart installation, you boot the system from a CD-ROM or an NFS-based miniroot. The installation software is loaded and run from this miniroot. Therefore, a Solaris Flash archive that you stored in the CD-ROM or NFS-based miniroot is accessible as a local file. Use the following syntax for the `archive_location` keyword.

```bash
archive_location local_file path/filename
```

- **path**
  The location of the archive. The path must be accessible to the system as a local file while the system is booted from the Solaris 9 Installation CD or from the Solaris 9 DVD. The system cannot access `/net` when it is booted from the Solaris 9 Installation CD or from the Solaris 9 DVD.

- **filename**
  The name of the Solaris Flash archive file.

**Examples:**

```bash
archive_location local_file /archives/usrarchive
archive_location local_file /archives/usrarchive
```

**backup_media Profile Keyword**

```bash
backup_media type path
```

**Note** – You can use `backup_media` only with the upgrade option when disk space reallocation is required.

`backup_media` defines the media that is to be used to back up file systems if space needs to be reallocated during an upgrade because of insufficient space. If multiple tapes or diskettes are required for the backup, you are prompted to insert tapes or diskettes during the upgrade.
Valid `type` Values | Valid `path` Values | Specification
---|---|---
local_tape | /dev/rmt/n | A local tape drive on the system that is being upgraded. `path` must be the character (raw) device path for the tape drive. `n` is the number of the tape drive.
local_diskette | /dev/rdiskette\n | A local diskette drive on the system that is being upgraded. `path` must be the character (raw) device path for the diskette drive. `n` is the number of the diskette drive.
local_filesystem | /dev/dsk/c\t\x20\x7fd\t\x20\x7d\y\x20\z\y/file_system | A local file system on the system that is being upgraded. You cannot specify a local file system that is being changed by the upgrade. `path` can be a block device path for a disk slice. For example, the `tx` in `/dev/dsk/c\t\x20\x7fd\t\x20\x7d\y\x20\z\y/file_system` might not be needed. Or, `path` can be the absolute path to a file system that is mounted by the `/etc/vfstab` file.
remote_filesystem | host:\x2f\file_system | An NFS file system on a remote system. `path` must include the name or IP address of the remote system, `host`, and the absolute path to the NFS file system, `file_system`. The NFS file system must have read/write access.
remote_system | user@host:\x2f\directory | A directory on a remote system that can be reached by a remote shell, `rsh`. The system that is being upgraded must have access to the remote system through the remote system’s `.rhosts` file. `path` must include the name of the remote system `host` and the absolute path to the directory `directory`. If a user login ID `user` is not specified, `root` is used by default.

Examples:
- backup_media local_tape /dev/rmt/0
- backup_media local_diskette /dev/rdiskette1
- backup_media local_filesystem /dev/dsk/c0t3d0s4
- backup_media local_filesystem /export
- backup_media remote_filesystem system1:/export/temp
- backup_media remote_system user1@system1:/export/temp
**boot_device** Profile Keyword

**boot_device device eeprom**

`boot_device` designates the device where the JumpStart program is to install the root (/) file system and the system’s boot device.

If you do not specify the `boot_device` keyword in a profile, the following `boot_device` keyword is specified by default during the installation: `boot_device any update`.

*device* – Use one of the following values.

- **SPARC**: `cwtxdysz` or `cxdysz` – The disk slice where the JumpStart program places the root (/) file system, for example, `c0t0d0s0`.
- **x86**: `cwtxdy` or `cxdy` – The disk where the JumpStart program places the root (/) file system, for example, `c0d0`.
- **existing** – The JumpStart program places the root (/) file system on the system’s existing boot device.
- **any** – The JumpStart program chooses where to place the root (/) file system. The JumpStart program attempts to use the system’s existing boot device. The JumpStart program might choose a different boot device if necessary.

*eeprom* – Choose to update or preserve the system’s EEPROM.

- **SPARC**: Choose if you want to update or preserve the system’s EEPROM to the specified boot device.
- **x86**: You must specify the `preserve` value.

  - **update** – The JumpStart program updates the system’s EEPROM to the specified boot device so that the installed system automatically boots from it.
  - **preserve** – The boot device value in the system’s EEPROM is not changed. If you specify a new boot device without changing the system’s EEPROM, you need to change the system’s EEPROM manually so it can automatically boot from the new boot device.

**Note** – SPARC: On SPARC systems, the `eeprom` value also allows you to update the system’s EEPROM if you change the system’s current boot device. By updating the system’s EEPROM, the system can automatically boot from the new boot device.

Example:

`boot_device c0t0d0s2 update`
**Note** – `boot_device` must match any `filesys` keywords that specify the root (`/`) file system and the `root_device` keyword, if specified.

---

**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 use the same platform group as the server. You must specify each platform group that you want the operating system server to support.

Valid values for `karch_value` are `sun4m`, `sun4u`, and `i86pc`. For a detailed list of platform names and various systems, see *Solaris 9 Sun Hardware Guide*.

**Note** – You can use `client_arch` only when `system_type` is specified as `server`.

---

**client_root** Profile Keyword

`client_root root_size`

`client_root` defines the amount of root space, `root_size` in Mbytes, to allocate for each client. If you do not specify `client_root` in a server’s profile, the installation software allocates 15 Mbytes of root space per client. The size of the client root area is used in combination with the `num_clients` keyword to determine how much space to reserve for the `/export/root` file system.

**Note** – You can use `client_root` only when `system_type` is specified as `server`.

---

**client_swap** Profile Keyword

`client_swap swap_size`

`client_swap` defines the amount of swap space, `swap_size` in Mbytes, to allocate for each diskless client. If you do not specify `client_swap` in the profile, 32 Mbytes of swap space is allocated by default.

Example:

```
client_swap 64
```
The example specifies that each diskless client is to have a swap space of 64 Mbytes.

**Note** – You can use `client_swap` only when `system_type` is specified as `server`.

*How the Size of swap Is Determined*

If a profile does not specify the size of `swap`, the JumpStart program determines the size of the swap space, based on the system’s physical memory. Table 28–5 shows how the size of `swap` is determined during a custom JumpStart installation.

**TABLE 28–5 Determining `swap` Size**

<table>
<thead>
<tr>
<th>Physical Memory (in Mbytes)</th>
<th>Swap Space (in Mbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16–64</td>
<td>32</td>
</tr>
<tr>
<td>64–128</td>
<td>64</td>
</tr>
<tr>
<td>128–512</td>
<td>128</td>
</tr>
<tr>
<td>Greater than 512</td>
<td>256</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Software Group</th>
<th><code>group_name</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Core System Support Software Group</td>
<td>SUNWCreq</td>
</tr>
<tr>
<td>End User Solaris Software Group</td>
<td>SUNWCuser</td>
</tr>
<tr>
<td>Developer Solaris Software Group</td>
<td>SUNWCprog</td>
</tr>
</tbody>
</table>
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)

```plaintext
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 `cxtyd` or `cyd`, for example, `c0t3d0`.

**Note** – You cannot specify the `dontuse` keyword and the `usedisk` keyword in the same profile.

### x86: `fdisk` Profile Keyword

```
fdisk  disk_name type size
```

`fdisk` defines how the `fdisk` partitions are set up on an x86 based system. You can specify `fdisk` more than once. When `fdisk` partitions an x86 based system, the following occurs:

- All `fdisk` partitions on the disk are preserved unless you delete the partitions with the `fdisk` keyword by assigning `size` the value of `delete` or `0`. Also, all existing `fdisk` partitions are deleted when `size` is set to `all`.
- A Solaris `fdisk` partition that contains a root (`/`) file system is always designated as the active partition on the disk.

**x86 only** – The system boots from the active partition by default.

- If the `fdisk` keyword is not specified in a profile, the following `fdisk` keyword is used by default during the installation:
  
  ```
  fdisk all solaris maxfree
  ```
- `fdisk` entries are processed in the order in which the entries are listed in the profile.

`disk_name` – Use the following values to specify where the `fdisk` partition is to be created or deleted:

- `cxtyd` or `cyd` – A specific disk, for example, `c0t3d0`.
- `rootdisk` – The variable that contains the value of the system’s root disk, which is determined by the JumpStart program as described in “How the System’s Root Disk Is Determined” on page 307.
- `all` – All the selected disks.

`type` – Use the following values to specify the type of `fdisk` partition that is to be created or deleted on the specified disk:

- `solaris` – A Solaris `fdisk` partition (SUNIXOS `fdisk` type).
- dosprimary – An alias for primary DOS fdisk partitions, not for fdisk partitions that are extended or reserved for data DOS. When you delete fdisk partitions by assigning size the value delete, dosprimary is an alias for the DOSHUGE, DOSOS12, and DOSOS16 fdisk types. When you create an fdisk partition, dosprimary is an alias for the DOSHUGE fdisk partition.

- DDD – An integer fdisk partition. DDD is an integer between 1 and 255 inclusive.

  x86 only – You can specify this value only if size is delete.

- 0xHH – A hexadecimal fdisk partition. HH is a hexadecimal number between 01 and FF.

  x86 only – You can specify this value only if size is delete.

The following table shows the integer and hexadecimal numbers for some of the fdisk types.

<table>
<thead>
<tr>
<th>fdisk Type</th>
<th>DDD</th>
<th>HH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSOS12</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>PCIXOS</td>
<td>2</td>
<td>02</td>
</tr>
<tr>
<td>DOSOS16</td>
<td>4</td>
<td>04</td>
</tr>
<tr>
<td>EXTDOS</td>
<td>5</td>
<td>05</td>
</tr>
<tr>
<td>DOSHUGE</td>
<td>6</td>
<td>06</td>
</tr>
<tr>
<td>DOSDATA</td>
<td>86</td>
<td>56</td>
</tr>
<tr>
<td>OTHEROS</td>
<td>98</td>
<td>62</td>
</tr>
<tr>
<td>UNIXOS</td>
<td>99</td>
<td>63</td>
</tr>
</tbody>
</table>

- size – Use one of the following values:

  - DDD – An fdisk partition of size DDD in Mbytes is created on the specified disk. DDD must be an integer, and the JumpStart program automatically rounds the number up to the nearest cylinder boundary. Specifying a value of 0 is the same as specifying delete.
  
  - all – An fdisk partition is created on the entire disk. All existing fdisk partitions are deleted.
**x86 only** – The all value can be specified only if `type` is `solaris`.

- **maxfree** – An `fdisk` partition is created in the largest contiguous free space on the specified disk. If an `fdisk` partition of the specified `type` already exists on the disk, the existing `fdisk` partition is used. A new `fdisk` partition is not created on the disk.

**x86 only** – The disk must contain at least one unused `fdisk` partition. Also, the disk must have free space or installation fails. The `maxfree` value can be specified only if `type` is `solaris` or `dosprimary`.

- **delete** – All `fdisk` partitions of the specified `type` are deleted on the specified disk.

**filesys** Profile Keyword (Mounting Remote File Systems)

`filesys server:path server_address mount_pt_name [mount_options]`

By using `filesys` with the listed values, the JumpStart program sets up the installed system to automatically mount remote file systems when the system boots. You can specify `filesys` more than once.

Example:

`filesys sherlock:/export/home/user2 - /home`

**server:** – The name of the server where the remote file system is located, followed by a colon.

**path** – The remote file system’s mount-point name. For example, `/usr` or `/export/home`.

**server_address** – The IP address of the server that is specified in `server:path`. If a name service is not running on the network, the `server_address` value can be used to populate the `/etc/hosts` file with the server’s host name and IP address. If you are not specifying the server’s IP address, you must specify a minus sign (`-`). For example, if you have a name service that is running on the network, you do not need to specify the server’s IP address.

**mount_pt_name** – The name of the mount point on which the remote file system is to be mounted.
mount_options – One or more mount options, which is the same as the -o option of the mount(1M) command. The mount options are added to the /etc/vfstab entry for the specified mount_pt_name.

Note – If you need to specify more than one mount option, the mount options must be separated by commas and no spaces (ro,quota for example).

filesys Profile Keyword (Creating Local File Systems)

filesys slice size [file_system optional_parameters]

By using filesys with the values that are listed, the JumpStart program creates local file systems during the installation. You can specify filesys more than once.

slice – Use one of the following values:

- any – The JumpStart program places the file system on any disk.

Note – You cannot specify any when size is existing, all, free, start:size, or ignore.

- cwtxdysz or cxdysz – 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 307. 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 *cwt.xdysz*. 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 *cwt.xdysz*.

- **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.

**forced_deployment** Profile Keyword (Installing Solaris Flash Differential Archives)

*forced_deployment*

*forced_deployment* forces the installation of a Solaris Flash differential archive onto a clone system that is different than the software expects.

Caution – If you use *forced_deployment*, all new files are deleted to bring the clone system to the expected state. If you are not certain that you want files deleted, use the default which protects new files by stopping the installation.

**geo** Profile Keyword

*geo locale*

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_Africa</td>
<td>Northern Africa, including Egypt</td>
</tr>
<tr>
<td>C_America</td>
<td>Central America, including Costa Rica, El Salvador, Guatemala, Mexico, Nicaragua, Panama</td>
</tr>
<tr>
<td>N_America</td>
<td>North America, including Canada, United States</td>
</tr>
<tr>
<td>S_America</td>
<td>South America, including Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Paraguay, Peru, Uruguay, Venezuela</td>
</tr>
<tr>
<td>Asia</td>
<td>Asia, including Japan, Republic of Korea, People’s Republic of China, Taiwan, Thailand</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ausi</td>
<td>Australasia, including Australia, New Zealand</td>
</tr>
<tr>
<td>C_Europe</td>
<td>Central Europe, including Austria, Czech Republic, Germany, Hungary, Poland, Slovakia, Switzerland</td>
</tr>
<tr>
<td>E_Europe</td>
<td>Eastern Europe, including Albania, Bosnia, Bulgaria, Croatia, Estonia, Latvia, Lithuania, Macedonia, Romania, Russia, Serbia, Slovenia, Turkey</td>
</tr>
<tr>
<td>N_Europe</td>
<td>Northern Europe, including Denmark, Finland, Iceland, Norway, Sweden</td>
</tr>
<tr>
<td>S_Europe</td>
<td>Southern Europe, including Greece, Italy, Portugal, Spain</td>
</tr>
<tr>
<td>W_Europe</td>
<td>Western Europe, including Belgium, France, Great Britain, Ireland, Netherlands</td>
</tr>
<tr>
<td>M_East</td>
<td>Middle East, including Israel</td>
</tr>
</tbody>
</table>

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 Solaris Flash archive on the system.

**Note** – You must specify `install_type` in a profile, and `install_type` must be the first profile keyword in every profile.

You must use one of the following options for the `initial_upgrade_flash_switch`:

- `initial_install` – Specifies to perform an initial installation of the Solaris operating environment
- `upgrade` – Specifies to perform an upgrade of the Solaris operating environment
- `flash_install` – Specifies to install a Solaris Flash archive that overwrites all files
- `flash_update` – Specifies to install a Solaris Flash differential archive that overwrites only the files that are specified
Note – Some profile keywords can only be used with the initial_install option. Some profile keywords can only be used with the upgrade option. Some profile keywords can only be used with the flash_install option.

**isa_bits** Profile Keyword

`isa_bits bit_switch`

`isa_bits` specifies whether 64-bit or 32-bit Solaris 9 packages are to be installed.

`bit_switch` represents the option 64 or 32, which you use to indicate whether 64-bit or 32-bit Solaris 9 packages are to be installed. If you do not set this keyword in the profile, the JumpStart program installs systems as follows:

- 64-bit packages on UltraSPARC™ systems
- 32-bit packages on all other systems

Note – If you use the `isa_bits` keyword, you must also use the latest check script in the `solaris_9/Misc/jumpstart_sample` directory on the Solaris 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 cwx dysz or cxdysz.

constraint – Use one of the following constraints for the specified file system:

- **changeable** – Auto-layout can move the file system to another location and it can change the file system size. The changeable constraint can only be specified on file systems that are mounted by the /etc/vfstab file. You can change the file system’s size by specifying the minimum_size value.

  When you mark a file system as changeable and minimum_size is not specified, the file system’s minimum size is set to 10 percent more than the minimum size that is required. For example, if the minimum size for a file system is 100 Mbytes, the changed size is 110 Mbytes. If minimum_size is specified, any free space that remains, original size minus minimum size, is used for other file systems.

- **movable** – Auto-layout can move the file system to another slice on the same disk or different disk. The file system size remains the same.

- **available** – Auto-layout can use all of the space on the file system to reallocate space. All of the data in the file system is lost. The available constraint can only be specified on file systems that are not mounted by the /etc/vfstab file.

- **collapse** – Auto-layout moves and collapses the specified file system into the parent file system. You can use the collapse option to reduce the number of file systems on a system as part of the upgrade. For example, if a system has the /usr and /usr/share file systems, collapsing the /usr/share file system moves the file system into /usr, the parent file system. You can specify the collapse constraint only on file systems that are mounted by the /etc/vfstab file.

minimum_size – Specifies the size of the file system after auto-layout reallocates space. The minimum_size option enables you to change the size of a file system. The size of the file system might be larger if unallocated space is added to the file system. But, the
size is never less than the value you specify. The *minimum_size* value is optional. Use this value only if you have marked a file system as changeable and the minimum size cannot be less than what the file system needs for the existing file system contents.

Examples:

```
layout_constraint c0t3d0s1 changeable 200
layout_constraint c0t3d0s4 movable
layout_constraint c0t3d1s3 available
layout_constraint c0t2d0s1 collapse
```

**local_customization** Profile Keyword (Installing Solaris Flash Archives)

```
local_customization local directory
```

Before you install a Solaris Flash archive on a clone system, you can create custom scripts to preserve local configurations on the clone system. The **local_customization** keyword designates the directory where you have stored these scripts. **local directory** is the path to the script on the clone system. For information on predeployment and postdeployment scripts, see “Creating Customization Scripts” on page 171.

**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 **locale** 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.
no_content_check Profile Keyword (Installing Solaris Flash Archives)

no_content_check

When installing a clone system with a Solaris Flash differential archive, you can use the no_content_check keyword to ignore file-by-file validation. File-by-file validation ensures that the clone system is a duplicate of the master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.

Caution – If you use no_content_check, all new files are deleted to bring the clone system to the expected state. If you are not certain that you want files deleted, use the default which protects new files by stopping the installation.

no_master_check Profile Keyword (Installing Solaris Flash Archives)

no_content_check

When installing a clone system with a Solaris Flash differential archive, you can use the no_master_check keyword to ignore checking the clone system to make sure it was built from the original master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system. For information on installing Solaris Flash differential archives, see “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” on page 257.

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.


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 `c0t0d0s2` or `cxdysz`.

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>
<thead>
<tr>
<th>Stage</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If the root_device keyword is specified in the profile, the JumpStart program sets rootdisk to the root device.</td>
</tr>
<tr>
<td>2</td>
<td>If rootdisk is not set and the boot_device keyword is specified in the profile, the JumpStart program sets rootdisk to the boot device.</td>
</tr>
<tr>
<td>3</td>
<td>If rootdisk is not set and a filesys create size / entry is specified in the profile, the JumpStart program sets rootdisk to the disk that is specified in the entry.</td>
</tr>
<tr>
<td>4</td>
<td>If rootdisk is not set and a rootdisk.size entry is specified in the profile, the JumpStart program searches the system’s disks in kernel probe order for an existing root file system on the specified slice. If a disk is found, the JumpStart program sets rootdisk to the found disk.</td>
</tr>
<tr>
<td>5</td>
<td>If rootdisk is not set and partitioning existing is specified in the profile, the JumpStart program searches the system’s disks in kernel probe order for an existing root file system. If a root file system is not found or more than one is found, an error occurs. If a root file system is found, the JumpStart program sets rootdisk to the found disk.</td>
</tr>
<tr>
<td>6</td>
<td>If rootdisk is not set, the JumpStart program sets rootdisk to the disk where the root (/) file system is installed.</td>
</tr>
</tbody>
</table>

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 *cydz* 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.
### Installation Environment Variables

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK_INPUT</td>
<td>The path to the rules file in the JumpStart directory, which is mounted on /tmp/install_config/rules.</td>
</tr>
<tr>
<td>HOME</td>
<td>The root's home directory during installation, which is /tmp/root.</td>
</tr>
<tr>
<td>PATH</td>
<td>The shell search path during installation, which is /sbin:/usr/sbin/install_d:/usr:/usr/sbin:/usr/bin.</td>
</tr>
<tr>
<td>SI_ARCH</td>
<td>The hardware architecture of the install client. The SI_ARCH variable is set when the arch keyword is used in the rules file.</td>
</tr>
<tr>
<td>SI_BEGIN</td>
<td>The name of the begin script, if one is used.</td>
</tr>
<tr>
<td>SI_CLASS</td>
<td>The name of the profile that is used to install the install client.</td>
</tr>
<tr>
<td>SI_CONFIG_DIR</td>
<td>The path to the JumpStart directory, which is mounted on /tmp/instal_config.</td>
</tr>
<tr>
<td>SI_CONFIG_FILE</td>
<td>The path to the rules file in the JumpStart directory, which is mounted on /tmp/install_config/rules.</td>
</tr>
<tr>
<td>SI_CONFIG_PROG</td>
<td>The rules file.</td>
</tr>
<tr>
<td>SI_CUSTOM_PROBES_FILE</td>
<td>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 277. You can also use the file to extend the list of default probe keywords that are described in “Probe Keywords and Values” on page 311.</td>
</tr>
<tr>
<td>SI_DISKLIST</td>
<td>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 307.</td>
</tr>
<tr>
<td>SI_DISKSIZES</td>
<td>A comma-separated list of disk sizes on the install client. The SI_DISKSIZES variable is set when the disksize keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SI_DOMAINNAME</td>
<td>The domain name. The SI_DOMAINNAME variable is set when the dommainname keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SI_FINISH</td>
<td>The name of the finish script, if one is used.</td>
</tr>
<tr>
<td>SI_HOSTADDRESS</td>
<td>The install client’s IP address.</td>
</tr>
<tr>
<td>SI_HOSTID</td>
<td>The install client’s Ethernet address.</td>
</tr>
<tr>
<td>SI_HOSTNAME</td>
<td>The install client’s host name. The SI_HOSTNAME variable is set when the hostname keyword is used and matched in the rules file.</td>
</tr>
</tbody>
</table>
### Table 28-7: Installation Environment Variables (Continued)

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI_INSTALLED</td>
<td>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.</td>
</tr>
<tr>
<td>SI_INST_OS</td>
<td>The name of the operating system. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.</td>
</tr>
<tr>
<td>SI_INST_VER</td>
<td>The version of the operating system. SI_INST_OS and SI_INST_VER are used to determine the value of SI_INSTALLED.</td>
</tr>
<tr>
<td>SI_KARCH</td>
<td>The install client’s kernel architecture. The SI_KARCH variable is set when the karch keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SI_MEMSIZE</td>
<td>The amount of physical memory on the install client. The SI_MEMSIZE variable is set when the memsize keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SI_MODEL</td>
<td>The install client’s model name. The SI_MODEL variable is set when the model keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SI_NETWORK</td>
<td>The install client’s network number. The SI_NETWORK variable is set when the network keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SI_NUMDISKS</td>
<td>The number of disks on an install client. The SI_NUMDISKS variable is set when the disksize keyword is used and matched in the rules file. The SI_NUMDISKS and SI_DISKLIST variables are used to determine the physical disk to use for the rootdisk. rootdisk is described in “How the System’s Root Disk Is Determined” on page 307.</td>
</tr>
<tr>
<td>SI_OSNAME</td>
<td>The operating system release on the Solaris 9 software image. For example, you can use the SI_OSNAME variable in a script if you are installing the Solaris software on systems that are based on the version of the operating system on the Solaris 9 DVD or the Solaris 9 Software 1 of 2 CD image.</td>
</tr>
<tr>
<td>SI_PROFILE</td>
<td>The path to the profile in the mounted JumpStart directory. The path is /tmp/install_config/profile_name. If you are creating a derived profile, SI_PROFILE is set to the /tmp/install.input file.</td>
</tr>
<tr>
<td>SI_ROOTDISK</td>
<td>The device name of the disk that is represented by the logical name rootdisk. The SI_ROOTDISK variable is set when the disksize or the installed keyword is set to rootdisk in the rules file.</td>
</tr>
<tr>
<td>SI_ROOTDISKSIZE</td>
<td>The size of the disk that is represented by the logical name rootdisk. The SI_ROOTDISKSIZE variable is set when the disksize or the installed keyword is set to rootdisk in the rules file.</td>
</tr>
<tr>
<td>SI_SYS_STATE</td>
<td>The /a/etc/.sysIDtool.state file. You can edit this file in a finish script to prevent the sysidroot program from prompting for a root password before the system reboots.</td>
</tr>
</tbody>
</table>
### TABLE 28–7 Installation Environment Variables (Continued)

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI_TOTALDISK</td>
<td>The total amount of disk space on the install client. The SI_TOTALDISK variable is set when the totaldisk keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SHELL</td>
<td>The default shell during installation, which is /sbin/sh.</td>
</tr>
<tr>
<td>TERM</td>
<td>The install client's terminal type.</td>
</tr>
<tr>
<td>TZ</td>
<td>The default time zone, as specified in the NIS or NIS+ name service.</td>
</tr>
</tbody>
</table>

#### 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

<table>
<thead>
<tr>
<th>Rule Keyword</th>
<th>Equivalent Probe Keyword</th>
<th>Description of Probe Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>arch</td>
<td>arch</td>
<td>Determines the kernel architecture, i386 or SPARC, and sets SI_ARCH.</td>
</tr>
<tr>
<td>disksize</td>
<td>disks</td>
<td>Returns the size of a system’s disks in Mbytes in kernel probe order, c0t3d0s0, c0t3d0s1, c0t4d0s0. disksize sets SI_DISKLIST, SI_DISKSIZES, SI_NUMDISKS, and SI_TOTALDISK.</td>
</tr>
<tr>
<td>domainname</td>
<td>domainname</td>
<td>Returns a system’s NIS or NIS+ domain name or blank and sets SI_DOMAINNAME. The domainname keyword returns the output of domainname(1M).</td>
</tr>
<tr>
<td>hostaddress</td>
<td>hostaddress</td>
<td>Returns a system’s IP address, the first address that is listed in the output of ifconfig(1M) -a that is not lo0, and sets SI_HOSTADDRESS.</td>
</tr>
<tr>
<td>hostname</td>
<td>hostname</td>
<td>Returns a system’s host name that is the output from uname(1) -n and sets SI_HOSTNAME.</td>
</tr>
<tr>
<td>installed</td>
<td>installed</td>
<td>Returns the version name of the Solaris operating environment that is installed on a system and sets SI_ROOTDISK and SI_INSTALLED. If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is SystemV.</td>
</tr>
<tr>
<td>Rule Keyword</td>
<td>Equivalent Probe Keyword</td>
<td>Description of Probe Keyword</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>karch</td>
<td>karch</td>
<td>Returns a system’s platform group, for example i86pc, sun4m, and sun4, and sets SI_KARCH. For a list of platform names, see Solaris 9 Sun Hardware Platform Guide.</td>
</tr>
<tr>
<td>memsize</td>
<td>memsize</td>
<td>Returns the size of physical memory on a system in Mbytes and sets SI_MEMSIZE.</td>
</tr>
<tr>
<td>model</td>
<td>model</td>
<td>Returns a system’s platform name and sets SI_MODEL. For a list of platform names, see the Solaris 9 Sun Hardware Platform Guide.</td>
</tr>
<tr>
<td>network</td>
<td>network</td>
<td>Returns a system’s network number, which the JumpStart program determines by performing a logical AND between the system’s IP address and the subnet mask. The system’s IP address and the subnet mask are extracted from the first address that is listed in the output of ifconfig(1M) -a that is not lo0. The network keyword sets SI_NETWORK.</td>
</tr>
<tr>
<td>osname</td>
<td>osname</td>
<td>Returns the version and operating system name of the Solaris operating environment that is found on a CD and sets SI_OSNNAME. If the JumpStart program finds a Solaris release but is unable to determine the version, the version that is returned is SystemV.</td>
</tr>
<tr>
<td>rootdisk</td>
<td>rootdisk</td>
<td>Returns the name and size in Mbytes of a system’s root disk and sets SI_ROOTDISK.</td>
</tr>
<tr>
<td>totaldisk</td>
<td>totaldisk</td>
<td>Returns the total disk space on a system (in Mbytes) and sets SI_TOTALDISK. The total disk space includes all of the operational disks that are attached to a system.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Provides overview information on the Solaris Live Upgrade process.</td>
</tr>
<tr>
<td>31</td>
<td>Provides information that you need to know before creating a boot environment.</td>
</tr>
<tr>
<td>32</td>
<td>Provides step-by-step instructions for installing Solaris Live Upgrade, using the menus, and creating a boot environment.</td>
</tr>
<tr>
<td>33</td>
<td>Provides step-by-step instructions for an operating system upgrade or installing a Solaris Flash archive on a boot environment, switching a boot environment to make it active, and recovering quickly from a failed upgrade.</td>
</tr>
<tr>
<td>34</td>
<td>Provides step-by-step instructions for maintaining a boot environment and viewing status.</td>
</tr>
<tr>
<td>35</td>
<td>Provides examples of a Solaris Live Upgrade.</td>
</tr>
<tr>
<td>36</td>
<td>Lists Solaris Live Upgrade commands.</td>
</tr>
</tbody>
</table>
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 Solaris Flash archive on a boot environment. The original system configuration remains fully functional and unaffected by the upgrade or installation of a Solaris 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 335.

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 352.
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 335.

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 `vfsstab` 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 350
- “To Create a Boot Environment and Merge File Systems (Command-Line Interface)” on page 349

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 341.

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 (/)

One hard disk

Two hard disks

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

**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

One hard disk

Two hard disks

/swap

root (/)
/usr
/opt
/export1

Upgrade

Copy

HDisk0

HDisk0

HDisk1

Swap

root (/)
/usr
/opt
/export1

Upgrade

Copy

Current release X
Critical file system root (/)

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

Shared 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.
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.
Rather than an upgrade, you can install a Solaris Flash archive on a boot environment. The Solaris Flash installation feature enables you to create a single reference installation of the Solaris operating environment on a system 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 Solaris Flash installation feature, see Chapter 16.
When you install the Solaris Flash archive on a system, the archive replaces all the files on the existing boot environment as an initial installation would. Figure 30–5 shows an installation of a Solaris Flash archive on an inactive boot environment. For procedures on installing a Solaris Flash archive, see “Installing Solaris Flash Archives on a Boot Environment” on page 372.

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 378.
Activating a Boot Environment

One hard disk

- hdisk0
  - root (/)
  - /swap
  - root (/)
  - /usr
  - /opt
  - /export1

Activate

Fallback if needed

Two hard disks

- hdisk0
  - root (/)
  - /swap
  - root (/)
  - /usr
  - /opt
  - /export1

Activate

Fallback if needed

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

- Inactive release X
  - Critical file systems root (/)

- Shared file system

**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 382.

Figure 30–7 shows the switch that is made when you reboot to fallback.
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 49. This chapter contains the following sections:

- “Solaris Live Upgrade System Requirements” on page 329
- “Solaris Live Upgrade Disk Space Requirements” on page 330
- “Managing Packages and Patches With Solaris Live Upgrade” on page 330
- “Guidelines for Selecting Slices for File Systems” on page 333
- “Using Solaris Live Upgrade From a Remote System” on page 336

Solaris Live Upgrade System Requirements

- From a SPARC based system, you can upgrade from the Solaris 2.6, Solaris 7, or Solaris 8 operating environment to the Solaris 8 or Solaris 9 operating environment.
- From an x86 based system, you can upgrade from the Solaris 7 or 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 339.

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 364 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>
<thead>
<tr>
<th>Solaris 2.6 Release</th>
<th>Solaris 7 Release</th>
<th>Solaris 8 Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNWadmap</td>
<td>SUNWadmap</td>
<td>SUNWadmap</td>
</tr>
<tr>
<td>SUNWadmfw</td>
<td>SUNWadmc</td>
<td>SUNWadmc</td>
</tr>
<tr>
<td>SUNWadmc</td>
<td>SUNWlibC</td>
<td>SUNWlibC</td>
</tr>
<tr>
<td>SUNWmfrun</td>
<td></td>
<td>SUNWbzip</td>
</tr>
<tr>
<td>SUNWloc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUNWlibC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 metadevice.
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>
<thead>
<tr>
<th>Product</th>
<th>Source Slice</th>
<th>If Target Is a Metadevice for a root (/) File System</th>
<th>If Target Is Not a root (/) File System, But Is for /usr, /var, or /opt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris Volume Manager</td>
<td>For the root (/) file system, the source can be a metadevice or physical slice.</td>
<td>The root (/) file system, must be either a stripe with only a single disk or a mirror on a single-disk stripe.</td>
<td>Other file systems can be either a physical slice or metadevice.</td>
</tr>
<tr>
<td>Veritas VxVM Volume Manager</td>
<td>For the root (/) file system, the source can be a volume or physical slice.</td>
<td>The root (/) file system, cannot be a VxVM volume; root (/) must be a physical slice.</td>
<td>Other file systems can be either a physical slice or volume.</td>
</tr>
</tbody>
</table>

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/cnumt/nunmsnum`
- 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 363 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 454

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.
9 releases.

For a description of shareable and critical file systems, see “Creating a Boot Environment Overview” on page 316.

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 337
- “Using Solaris Live Upgrade Menus” on page 338
- “Task Map: Installing Solaris Live Upgrade and Creating Boot Environments” on page 339
- “Installing Solaris Live Upgrade” on page 339
- “Starting and Stopping Solaris Live Upgrade (Character Interface)” on page 340
- “Creating a New Boot Environment” on page 341

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

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install Solaris Live Upgrade packages</td>
<td>Install packages on your Solaris 2.6, Solaris 7, or Solaris 8 operating environment</td>
<td>“Installing Solaris Live Upgrade” on page 339</td>
</tr>
<tr>
<td>Start Solaris Live Upgrade</td>
<td>Start the Solaris Live Upgrade main menu</td>
<td>“Starting and Stopping Solaris Live Upgrade (Character Interface)” on page 340</td>
</tr>
<tr>
<td>Create a boot environment</td>
<td>Copy and reconfigure file systems to an inactive boot environment</td>
<td>“Creating a New Boot Environment” on page 341</td>
</tr>
</tbody>
</table>

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.
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.
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 316.

This section contains the following procedures:

- “To Create a Boot Environment (Character Interface)” on page 342
- “To Create a Boot Environment for the First Time (Command-Line Interface)” on page 347
- “To Create a Boot Environment and Merge File Systems (Command-Line Interface)” on page 349
- “To Create a Boot Environment and Split File Systems (Command-Line Interface)” on page 350
- “To Create a Boot Environment and Reconfigure Swap (Command-Line Interface)” on page 352
- “To Create a Boot Environment and Reconfigure Swap Using a List (Command-Line Interface)” on page 354
- “To Create a Boot Environment and Copy a Shareable File System (Command-Line Interface)” on page 356
- “To Create a Boot Environment From a Different Source (Command-Line Interface)” on page 357
- “To Create a Boot Environment for a Flash Archive (Command-Line Interface)” on page 358

▼ 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/cnumdnumtumsnum.
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/dnum.
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/volume_name.

- The 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 316

- 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 333.
  - 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.

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.
   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:

```bash
# lucreate [-A 'BE_description'] -c BE_name \n-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.
-c BE_name Assigns the name BE_name to the active boot environment. This option is required only when the first boot environment is created. If you run lucreate for the first time and you omit -c, you are prompted to name the active boot environment. If you use the -c option after the first boot environment creation, you receive an error message.

-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/cnumdnumtnums
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/dnum
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/volume_name
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- fs_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 32-1 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
   ```

   `-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` 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/cnumdnunt:numsnum
     - The name of a Solaris Volume Manager metadevice, of the form `/dev/md/dsk/dnum
     - The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/volume_name
     - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
   - `fs_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.
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.
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:device:fs_type

[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 metadevice, of the form /dev/md/dsk/dnum
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/volume_name
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent

- fs_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 32-3 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 -e 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.
-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 metadevice, of the form /dev/md/dsk/dnum
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/volume_name
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- fs_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.

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 354.

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. lucheate 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:

   ```
   # lucheate [-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.
-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 metadevice, of the form /dev/md/dsk/dnum
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/volume_name
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- fs_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).

-M slice_list

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

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

-n BE_name

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

EXAMPLE 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` 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/cnumdnumtnumsnunum`
     - The name of a Solaris Volume Manager metadevice, of the form `/dev/md/dsk/dnum`
     - The name of a Veritas Volume Manager volume, of the form `/dev/vx/dsk/volume_name`
     - The keyword `merged`, indicating that the file system at the specified mount point is to be merged with its parent
   - `fs_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.
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

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/cnumdnumtnum
  - The name of a Solaris Volume Manager metadevice, of the form 
    /dev/md/dsk/dnum
  - The name of a Veritas Volume Manager volume, of the form 
    /dev/vx/dsk/volume_name
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent
- fs_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

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

-s - Specifies that an empty boot environment be created.

-m mountpoint:device:fs_type 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/cnumdnumtnumsnums
  - The name of a Solaris Volume Manager metadevice, of the form /dev/md/dsk/dnum
  - The name of a Veritas Volume Manager volume, of the form /dev/vx/dsk/volume_name
  - The keyword merged, indicating that the file system at the specified mount point is to be merged with its parent

- fs_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 32-8 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 362
- “Upgrading a Boot Environment” on page 362
- “Installing Solaris Flash Archives on a Boot Environment” on page 372
- “Activating a Boot Environment” on page 378
- “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 382

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

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 366.
When you upgrade a boot environment with the latest operating environment, you do not affect the active boot environment. The new files merge with the inactive boot environment critical file systems, but shareable file systems are not changed.

Rather than upgrading, if you have created a Solaris Flash archive, you could install the archive on an inactive boot environment. The new files overwrite the inactive boot environment critical file systems, but shareable file systems are not changed. See “Installing Solaris Flash Archives on a Boot Environment” on page 372.

Upgrading Metadevices and Volumes

To upgrade or install an 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 an archive fails. To upgrade or install an 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 an 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/md/dsk/d10:ufs \
   -m /usr:/dev/dsk/c0t4d0s3:ufs -m /var:/dev/dsk/c0t4d0s4:ufs
```

You would then be able to upgrade or install an 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 an 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 Solaris 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 Solaris 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 Solaris Flash archive begins with creating a master system. After you have created a master system, add any patches and packages that you want to install. Then, create a Solaris Flash archive of the master system. Use Solaris Live Upgrade to install the archive on the new boot environment. You can copy the boot environment, change it, and distribute it as many times as necessary. For details about how to create a Solaris Flash archive, see Chapter 18. For information on using Solaris Live Upgrade to install a Solaris Flash archive, see “Installing Solaris Flash Archives on a Boot Environment” on page 372.
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, `lupgrade(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 366.

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.

<table>
<thead>
<tr>
<th>Installation Media Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network File System</td>
<td>Specify the path to the network file system where the installation image is located.</td>
</tr>
<tr>
<td>Local file</td>
<td>Specify the path to the local file system where the installation image is located.</td>
</tr>
<tr>
<td>Local tape</td>
<td>Specify the local tape device and the position on the tape where the installation image is located.</td>
</tr>
<tr>
<td>Local device, DVD, or CD</td>
<td>Specify the local device and the path to the installation image.</td>
</tr>
</tbody>
</table>

- 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 366

1. Log in as superuser.

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

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

   - `u` Upgrades an operating system image on a boot environment
   - `n BE_name` Specifies the name of the boot environment that is to be upgraded
   - `s os_image_path` Specifies the path name of a directory that contains an operating system image

   **EXAMPLE 33-1** Upgrading an Operating System Image on a Boot Environment (Command-Line Interface)

   In this example, the second_disk boot environment is upgraded.

   `# lupgrade -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:

   `# lupgrade -u -n BE_name -s os_image_path`
-u               Upgrades an operating system image on a boot environment
-n BE_name       Specifies the name of the boot environment that is to be upgraded
-s os_image_path Specifies the path name of a directory that contains an operating system image

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

4. This step is identical to the previous step, but the -u option is replaced by the -i option. Also, choose to run the installer on the second CD with menus or with text.
   - This command runs the installer on the second CD with menus.
     # luupgrade -i -n BE_name -s os_image_path
   - This command runs the installer on the second CD with text and requires no user interaction.
     # luupgrade -i -n BE_name -s os_image_path -O '-nodosplay -noconsole'

- i               Looks for an installation program on the specified medium and runs that program. The installer program is specified with -s.
-n BE_name        Specifies the name of the boot environment that is to be upgraded.
-s os_image_path  Specifies the path name of a directory that contains an operating system image.
-O '-nodosplay    (Optional) Runs the installer on the second CD in text mode and
-noconsole'       requires no user interaction.

The boot environment is ready to be activated. See “Activating a Boot Environment” on page 378

EXAMPLE 33-2 SPARC: Upgrading an Operating System Image From Multiple CDs
(Command-Line Interface)

In this example, the second_disk boot environment is upgraded and the installation image is on two CDs: the Solaris 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 -nodosplay 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 SPARC: 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.

▼ To Create a Profile to be Used by Solaris Live Upgrade

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

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

- For an upgrade with a profile, see “To Upgrade With a Profile by Using Solaris
  Live Upgrade (Command-Line Interface)” on page 371.
- For a Solaris Flash installation with a profile, see “To Install a Solaris Flash Archive
  With a Profile (Command-Line Interface)” on page 376

1. Use a text editor to create a text file.
   Name the file descriptively. Ensure that the name of the profile reflects how you
   intend to use the profile to install the Solaris software on a system. For example, you
   might name this profile upgrade_solaris9.

2. Add profile keywords and values to the profile.
   Only the following upgrade keywords can be used in a Solaris Live Upgrade profile.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Required) Install_type</td>
<td>Defines whether to upgrade the existing Solaris environment on a system or install a Solaris Flash archive on the system. The values are upgrade or flash_install.</td>
</tr>
<tr>
<td>(Optional) archive_location</td>
<td>Retrieves a Solaris Flash archive from a designated location. For a list of values that can be used with this keyword, see “archive_location Keyword” on page 283</td>
</tr>
</tbody>
</table>
TABLE 33–2 Live Upgrade Profile Keywords and Descriptions  (Continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) cluster (adding or deleting clusters)</td>
<td>Designates whether a cluster is to be added or deleted from the software group that is to be installed on the system. For a list of values that can be used with this keyword, see “cluster Profile Keyword (Adding Software Groups)” on page 292.</td>
</tr>
<tr>
<td>(Optional) geo</td>
<td>Designates the regional locale or locales that you want to install on a system or to add when upgrading a system. For a list of values that can be used with this keyword, see “geo Profile Keyword” on page 299.</td>
</tr>
<tr>
<td>(Optional) isa_bits</td>
<td>Specifies whether 64-bit or 32-bit Solaris 9 packages are to be installed. For a list of values that can be used with this keyword, see “isa bits Profile Keyword” on page 301.</td>
</tr>
<tr>
<td>(Optional) locale</td>
<td>Designates the locale packages you want to install or add when upgrading. For a list of values that can be used with this keyword, see “locale Profile Keyword” on page 303.</td>
</tr>
<tr>
<td>(Optional) package</td>
<td>Designates whether a package is to be added to or deleted from the software group that is to be installed on the system. For a list of values that can be used with this keyword, see “package Profile Keyword” on page 304.</td>
</tr>
</tbody>
</table>

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

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

5. Test the profile (optional).
   For a procedure to test the profile, see “To Test a Profile to Be Used By Solaris Live Upgrade” on page 370.

EXAMPLE 33–3 Creating a Profile to Be Used With Solaris Live Upgrade

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

```
# profile keywords                  profile values
# -----------------                -------------------------------
install_type                   upgrade
```
package  SUNWxwman  add
cluster  SUNWCacc  add
go  C_Europe
locale  zh_TW
locale  zh_TW.BIG5
locale  zh_TW.UTF-8
locale  zh_HK.UTF-8
locale  zh_HK.BIG5HK
locale  zh
locale  zh_CN.GB18030
locale  zh_CN.GBK
locale  zh_CN.UTF-8

To Test a Profile to Be Used By Solaris Live Upgrade

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

Test the profile.

```
# luupgrade -u -n BE_name -D -s os_image_path -j profile_path
```

- **-u**
  Upgrades an operating system image on a boot environment.

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

- **-D**
  luupgrade command uses the selected boot environment’s disk configuration to test the profile options passed with the -j option.

- **-s os_image_path**
  Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.

- **-j profile_path**
  Path to a profile that is configured for an upgrade. The profile must be in a directory on the local machine.

EXAMPLE 34-4 Testing a Profile by Using Solaris Live Upgrade

In the following example, the profile is named upgrade_solaris9, The profile is successfully tested on the inactive boot environment named, s9ulb08.
EXAMPLE 33–4 Testing a Profile by Using Solaris Live Upgrade  
(Continued)

# luupgrade -u -n s9u1b08 -D -s /net/installsvr/export/s9u1/combined.s9s_ulwos 
- j /var/tmp/flash_profile
Validating the contents of the media /net/installsvr/export/s9u1/combined.s9s_ulwos.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains Solaris version 9.
Locating upgrade profile template to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Live Upgrade requests.
Creating upgrade profile for BE s9u1b08.
Determining packages to install or upgrade for BE s9u1b08.
Simulating the operating system upgrade of the BE s9u1b08.
The operating system upgrade simulation is complete.
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of the
upgrade operation.
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of
cleanup operations required.
The Solaris upgrade of the boot environment s9u1b08 is complete.

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

▼ To Upgrade With a Profile by Using Solaris Live Upgrade (Command-Line Interface)

This procedure provides step-by-step instructions for upgrading a operating
environment by using a profile.

If you want to install a Solaris Flash archive by using a profile, see “To Install a Solaris
Flash Archive With a Profile (Command-Line Interface)” on page 376

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

1. Log in as superuser.

2. Create a profile.
   See Table 33–2 for a list of upgrade keywords that can be used in a Solaris Live
   Upgrade profile.

3. Type:

   # luupgrade -u -n BE_name -s os_image_path -j profile_path

   -u  Upgrades an operating system image on a boot environment

   Chapter 33 • Upgrading With Solaris Live Upgrade (Tasks)  371
- `n BE_name` Specifies the name of the boot environment that is to be upgraded.

- `s os_image_path` Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.

- `j profile_path` Path to a profile. The profile must be in a directory on the local machine. For information on creating a profile, see “To Create a Profile to be Used by Solaris Live Upgrade” on page 368.

The boot environment is ready to be activated.

**EXAMPLE 33-5 Upgrading a Boot Environment By Using a Custom JumpStart Profile (Command-Line Interface)**

In this example, an archive is installed on the `second_disk` boot environment. The `-j` option is used to access the profile. The boot environment is then ready to be activated. To create a profile, see “To Create a Profile to be Used by Solaris Live Upgrade” on page 368.

```
# luupgrade -u -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

---

## Installing Solaris Flash Archives on a Boot Environment

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

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

**Note** – For examples of the correct syntax for paths that are associated with archive storage, see “archive_location Keyword” on page 283.

Installing a Solaris Flash archive overwrites all files on the new boot environment except for shared files.
To use the Solaris Flash installation feature, you need to have installed the master system and to have created the Solaris Flash archive. For more information about Solaris Flash, see Chapter 16.

Upgrading Metadevices and Volumes

To upgrade or install a Solaris 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 archive fails. To upgrade or install an 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 an 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 archive on the new boot environment, activate it, and then re-mirror or encapsulate it manually.

If you choose to manually tear down 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 an archive.

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

1. From the Solaris Live Upgrade main menu, select Flash.
   The Flash an Inactive Boot Environment menu is displayed.

2. Type the name of the boot environment where you want to install the Solaris Flash archive and the location of the installation media:
   Name of Boot Environment: solaris_9
   Package media: /net/install-svr/export/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.
   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 Solaris 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
     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 Solaris Flash archive.
<table>
<thead>
<tr>
<th>Media Selected</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Specify the URL and proxy information that is needed to access the Solaris Flash archive.</td>
</tr>
<tr>
<td>NFS</td>
<td>Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.</td>
</tr>
<tr>
<td>Local file</td>
<td>Specify the path to the local file system where the Solaris Flash archive is located.</td>
</tr>
<tr>
<td>Local tape</td>
<td>Specify the local tape device and the position on the tape where the Solaris Flash archive is located.</td>
</tr>
<tr>
<td>Local device</td>
<td>Specify the local device, the path to the Solaris Flash archive, and the type of file system on which the Solaris Flash archive is located.</td>
</tr>
</tbody>
</table>

A Retrieval submenu is displayed, similar to the following example, which depends on the media you selected.

**NFS Location:**

1. Type the path to the archive, as in the following example.

   **NFS Location:** host:/path/to archive.flar

2. Press F3 to add the archive to the list.

3. When the list contains the archives that you want to install, press F6 to exit.

4. Press F3 to install one or more archives.

   The Solaris Flash archive is installed on the boot environment. All files on the boot environment are overwritten, except for shareable files.

   The boot environment is ready for activation. See “To Activate a Boot Environment (Character Interface)” on page 379.

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

1. Log in as superuser.

2. Type:

   ```bash
   # luupgrade -f -n BE_name -s os_image_path -a archive
   ```
-f Indicates to install an operating system from a Solaris Flash archive.

-n BE_name Specifies the name of the boot environment that is to be installed with an archive.

-s os_image_path Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.

-a archive Path to the Solaris Flash archive when the archive is available on the local file system. The operating system image versions specified with the -s option and the -a option must be identical.

EXAMPLE 33-6 Installing Solaris Flash Archives on a Boot Environment (Command-Line Interface)

In this example, an archive is installed on the second_disk boot environment. The archive is located on the local system. The operating systems versions for the -s and -a options are both Solaris 9 releases. All files are overwritten on second_disk except shareable files.

```
# lupgrade -f -n second_disk \
-s /net/installmachine/export/solaris9/OS_image \
-a /net/server/archive/solaris9
```

The boot environment is ready to be activated.

▼ To Install a Solaris Flash Archive With a Profile (Command-Line Interface)

This procedures provides the steps to install a Solaris Flash archive by using a profile.

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

1. Log in as superuser.

2. Create a profile.
   See Table 33–2 for a list of keywords that can be used in a Solaris Live Upgrade profile.

3. Type:
   ```
   # lupgrade -f -n BE_name -s os_image_path -j profile_path
   ```

   -f Indicates to install an operating system from a Solaris Flash archive.
The boot environment is ready to be activated.

**EXAMPLE 33-7 Upgrading a Boot Environment With a Profile (Command-Line Interface)**

In this example, an archive is installed on the `second_disk` boot environment. The `-j` option is used to access the profile. The boot environment is then ready to be activated. To create a profile, see “To Create a Profile to be Used by Solaris Live Upgrade” on page 368.

```bash
# luupgrade -f -n second_disk \
- s /net/installmachine/export/solarisX/OS_image \ 
- j /var/tmp/profile
```

▼ **To Install a Solaris Flash Archive With a Profile Keyword (Command-Line Interface)**

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

1. Log in as superuser.
2. Type:

```bash
# luupgrade -f -n BE_name -s os_image_path -j 'profile'
```

- `-f` Indicates to upgrade an operating system from a Solaris Flash archive.
- `-n BE_name` Specifies the name of the boot environment that is to be upgraded.
- `-s os_image_path` Specifies the path name of a directory that contains an operating system image. This directory can be on an installation medium, such as a DVD-ROM, CD-ROM, or it can be an NFS or UFS directory.
The boot environment is ready to be activated.

EXAMPLE 33-8 Installing a Solaris Flash Archive By Using a Profile (Command-Line Interface)

In this example, an archive is installed on the second_disk boot environment. The -J option and the archive_location keyword is used to retrieve the archive. All files are overwritten on second_disk except shareable files.

```
# luupgrade -f -n second_disk \  
  -s /net/installmachine/export/solarisX/OS_image \  
  -J ‘archive_location http://example.com/myflash.flar’
```

Activating a Boot Environment

Activating a boot environment makes it bootable on the next reboot of the system. You can also switch back quickly to the original boot environment if a failure occurs on booting the newly active boot environment. See “Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)” on page 382

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 391.
- If the boot environment is not the current boot environment, you cannot have mounted the partitions of that boot environment by using lmount(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 395.

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 352.
x86: (Optional) Update Boot Diskette Before Activating

If you’re not using the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette to boot the system, you can skip this procedure. If you use the Device Configuration Assistant to boot, you need to update the boot diskette. This procedure matches a boot diskette to your release by overwriting the existing diskette or writing to a new diskette.

1. Insert the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette. This is either the existing diskette that is overwritten or a new diskette.

2. Update the boot diskette with the latest image for this release.

3. Eject the boot diskette.

4. Type:

```
volcheck
```

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

```
cp /a/boot/solaris/bootenv.rc /floppy/floppy0/solaris/bootenv.rc
```

6. Check the input-device and output-device on the diskette to ensure that they are correct. If not, then update them.

You are ready to activate the new boot environment.

To Activate a Boot Environment (Character Interface)

The first time you boot from a newly created boot environment, Solaris Live Upgrade software synchronizes the new boot environment with the boot environment that was last active. 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
   ```
3. **Reboot.**

```bash
# 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 33–9** Activating a Boot Environment (Command-Line Interface)

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

```bash
# /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:

   ```bash
   # /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.

   ```bash
   # init 6
   ```

**EXAMPLE 33-10 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.

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

---

**Failure Recovery: Falling Back to the Original Boot Environment (Command-Line Interface)**

If a failure is detected after upgrading or if the application is not compatible with an upgraded component, fall back to the original boot environment by using one of the following procedures, depending on your platform. For SPARC based systems, the first procedure runs `luactivate` to switch back to the original boot environment. If
the first procedure fails, use the second procedure to boot from a DVD-ROM, CD-ROM, or net image. For x86 based systems, choose the procedure that is based on where the root (/) file systems are mounted, on the same or different physical disks.

▼ To Fall Back Despite Successful New Boot Environment Creation

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

1. Log in as superuser.
2. Type:
   
   ```
   # /usr/sbin/luactivate BE_name
   ```

   Where `BE_name` specifies the name of the boot environment to be activated

3. Reboot.

   ```
   # init 6
   ```

   The previous working boot environment becomes the active boot environment.

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

- If you experience a failure while booting the new boot environment and can boot the original boot environment in single-user mode, use this procedure to fall back to the original boot environment.
- If you need to boot from media or a net installation image, see “SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 384

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 “SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 384.
   - If the prompt is displayed, continue.

3. At the prompt, type:

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

   A message displays that the fallback activation is successful.

4. Reboot.

   # init 6

   The previous working boot environment becomes the active boot environment.

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

Use this procedure to boot from a DVD, CD, or a net installation image. You need to mount the root (/) slice from the last active boot environment. Then run the luactivate command, which makes the switch. When you reboot, the last active boot environment is up and running again.

1. At the OK prompt, boot the machine to single-user state from the Solaris 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/cnumtnumtnumnumnum.

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/cnumtnumtnumnumnum.

4. From the active boot environment root (/) slice, type:

   # /mnt/sbin/luactivate

   luactivate activates the previous working boot environment and indicates the
result.

5. Unmount /mnt

   # umount /mnt

6. Reboot.

   # init 6

   The previous working boot environment becomes the active boot environment.

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

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

1. Reboot the machine and enter the appropriate BIOS menus.
If your boot devices are SCSI, refer to documentation on your SCSI controller on how to enter the SCSI BIOS.

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

2. Follow the appropriate BIOS documentation to change the boot device back to the original boot environment’s boot device if different.

3. Save the BIOS changes.

4. Exit BIOS to begin the boot process.

5. Type `b - s` to boot the machine to single-user state.

6. Type:
   ```
   # /sbin/luactivate
   ```

7. Reboot.
   ```
   # init 6
   ```

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

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

1. Decide how to boot the system.
   - If you boot from the Solaris 9 DVD or the Solaris 9 Installation CD, insert the disc. Your system’s BIOS must support booting from a DVD or CD.
   - If you boot from the network, use Preboot Execution Environment (PXE) network boot. The system must support PXE. Enable the system to use PXE by using the systems’s BIOS setup tool or the network adapter’s configuration setup tool.
   - If you boot from a diskette, insert Solaris 9 Device Configuration Assistant x86 Platform Edition diskette into the system’s diskette drive.
**x86 only** – You can copy the Device Configuration Assistant software to a diskette by the following methods:

- Copy the software to a diskette from the Solaris 9 x86 Platform Edition DVD or Solaris 9 Software 2 of 2 x86 Platform Edition CD by using this procedure: Appendix E.
- Download and copy the software to a diskette from the Solaris Developer Connection at http://soldc.sun.com/support/drivers/dca_diskettes.

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

2. **Type** `b -s` **to boot the machine to single-user state.**

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

   ```
   # fsck mount_point
   
   mount_point
   ``

   A root file system that is known and reliable

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

   ```
   # mount device_name /mnt/sbin
   
   device_name
   ``

   Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is entered in the form of `/dev/dsk/cnum/dnum/snum/dnum`.

5. **From the active boot environment root slice, type:**

   ```
   # /mnt/sbin/luactivate
   
   luactivate activates the previous working boot environment and indicates the result.
   
   # umount device_name /mnt/sbin
   
   Unmount /mnt/sbin.**
device_name  Specifies the location of the root (/) file system on the disk device of the boot environment you want to fall back to. The device name is entered in the form of /dev/dsk/cnumtnumdnunnum.

7. 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 390
- “Displaying the Status of All Boot Environments” on page 391
- “Updating a Previously Configured Boot Environment” on page 392
- “Canceling a Scheduled Create, Upgrade, or Copy Job” on page 394
- “Comparing Boot Environments” on page 395
- “Deleting an Inactive Boot Environment” on page 397
- “Displaying the Name of the Active Boot Environment” on page 398
- “Changing the Name of a Boot Environment” on page 399
- “Adding or Changing a Description Associated With a Boot Environment Name” on page 400
- “Viewing the Configuration of a Boot Environment” on page 403


## Overview of Solaris Live Upgrade Maintenance

### TABLE 34-1 Overview of Solaris Live Upgrade Maintenance

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) View Status</td>
<td>View whether a boot environment is active, being activated, scheduled to be activated, or in the midst of a comparison</td>
<td>“Displaying the Status of All Boot Environments” on page 391</td>
</tr>
<tr>
<td></td>
<td>Compare the active and inactive boot environments</td>
<td>“Comparing Boot Environments” on page 395</td>
</tr>
<tr>
<td></td>
<td>Display the name of the active boot environment</td>
<td>“Displaying the Name of the Active Boot Environment” on page 398</td>
</tr>
<tr>
<td></td>
<td>View the configurations of a boot environment</td>
<td>“Viewing the Configuration of a Boot Environment” on page 403</td>
</tr>
<tr>
<td>(Optional) Update an inactive boot environment</td>
<td>Copy file systems from the active boot environment again without changing the configuration of file systems</td>
<td>“Updating a Previously Configured Boot Environment” on page 392</td>
</tr>
<tr>
<td>(Optional) Other tasks</td>
<td>Delete a boot environment</td>
<td>“Deleting an Inactive Boot Environment” on page 397</td>
</tr>
<tr>
<td></td>
<td>Change the name of a boot environment</td>
<td>“Changing the Name of a Boot Environment” on page 399</td>
</tr>
<tr>
<td></td>
<td>Add or change a description that is associated with a boot environment name</td>
<td>“Adding or Changing a Description Associated With a Boot Environment Name” on page 400</td>
</tr>
<tr>
<td></td>
<td>Cancel scheduled jobs</td>
<td>“Canceling a Scheduled Create, Upgrade, or Copy Job” on page 394</td>
</tr>
</tbody>
</table>
Displaying the Status of All Boot Environments

Use the Status menu or the `lstatus` 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:

<table>
<thead>
<tr>
<th>BE_name</th>
<th>Complete</th>
<th>Active</th>
<th>ActiveOnReboot</th>
<th>CopyStatus</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk_a_S7</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
</tr>
<tr>
<td>disk_b_S7database</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>COPYING</td>
</tr>
<tr>
<td>disk_b_S8</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>-</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>BE_name</th>
<th>Complete</th>
<th>Active</th>
<th>ActiveOnReboot</th>
<th>CopyStatus</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk_a_S7</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
</tr>
<tr>
<td>disk_b_S7database</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>COPYING</td>
</tr>
<tr>
<td>disk_b_S8</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>-</td>
</tr>
</tbody>
</table>
   ```

**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 391 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 394.
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 394.

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.
-a source_BE  (Optional) Specifies the name of the source boot environment that contains the file systems to be copied to the target boot environment. If you omit this option, lumake uses the current boot environment as the source.

-t time  (Optional) Set up a batch job to copy over file systems on a specified boot environment at a specified time. The time is given in the format that is specified by the man page, at(1).

-m email_address  (Optional) Enables you to email lumake output to a specified address on command completion. email_address is not checked. You can use this option only in conjunction with -t.

EXAMPLE 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 394.

## 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 391.

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.


4. Type the names of the original (active) boot environment, the inactive boot environment, and the path to a file:
   ```
   Name of Parent: solaris8
   Name of Child: solaris8-1
   Full Pathname of the file to Store Output: /tmp/compare
   ```

5. To save to the file, press F3.
   The Compare menu displays the following file attributes:
   - Mode.
   - Number of links.
   - Owner.
Group.

Checksum – Computes checksums only if the file in the specified boot environment matches its counterpart on the active boot environment in all of the fields that are described previously. If everything matches but the checksums differ, the differing checksums are appended to the entries for the compared files.

Size.

Existence of files in only one boot environment.

6. To return to the Compare menu, press F3.

To Compare Boot Environments (Command-Line Interface)

1. Log in as superuser.

2. Type:

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

   -i infile
   Compare files that are listed in infile. The files to be compared should be an absolute file name. If the entry in the file is a directory, then comparison is recursive to the directory. Use either this option or -t, not both.

   -t
   Compare only nonbinary files. This comparison uses the file(1) command on each file to determine if the file is a text file. Use either this option or -i, not both.

   -o outfile
   Redirect the output of differences to outfile.

   BE_name
   Specifies the name of the boot environment that is compared to the active boot environment.

EXAMPLE 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 391 to determine a boot environment’s status. Also, you cannot delete a boot environment that has file systems mounted with `lmount`.

▼ To Delete an Inactive Boot Environment (Character Interface)

1. From the main menu, select Delete.

2. Type the name of the inactive boot environment you want to delete:

   Name of boot environment: `solaris8`

   The inactive boot environment is deleted.

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

1. Log in as superuser.

2. Type:

   ```
   # ludelete BE_name
   ```

   `BE_name` specifies the name of the inactive boot environment that is to be deleted.

**EXAMPLE 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 391 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 391 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 347. 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
```

- `-n BE_name` Specifies the boot environment name and the new description to be associated with the name
- `-f file_name` 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'
```

- `-A 'BE_description'` 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)

```bash
#/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:
   ```bash
   # /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.

```bash
#/usr/sbin/ludesc -f rose.gif
second_disk
```

▼ To Determine a Boot Environment Description From a Name

1. Log in as superuser.
2. Type:
   ```bash
   # /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.

```bash
#/usr/sbin/ludesc -n second_disk Solaris 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

   
   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
   ---------------------------
<table>
<thead>
<tr>
<th>Device Path</th>
<th>Filesystem Type</th>
<th>Size</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/dsk/c0t0d0s1</td>
<td>swap</td>
<td>512.11</td>
<td>-</td>
</tr>
<tr>
<td>/dev/dsk/c0t4d0s3</td>
<td>ufs</td>
<td>3738.29</td>
<td>/</td>
</tr>
<tr>
<td>/dev/dsk/c0t4d0s4</td>
<td>ufs</td>
<td>510.24</td>
<td>/opt</td>
</tr>
</tbody>
</table>
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.

```bash
#/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.

```bash
# 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.

```bash
# 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.

```bash
# lustatus
```

<table>
<thead>
<tr>
<th>BE_name</th>
<th>Complete</th>
<th>Active</th>
<th>ActiveOnReboot</th>
<th>CopyStatus</th>
</tr>
</thead>
</table>

406 Solaris 9 Installation Guide • April 2003
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 407.
- The activation fails and you can boot back to the original boot environment. See “SPARC: To Fall Back From a Failed Boot Environment Activation” on page 407.
- The activation fails and you must boot back to the original boot environment by using media or a net installation image. See “SPARC: To Fall Back to the Original Boot Environment by Using a DVD, CD, or Net Installation Image” on page 408.

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
```

SPARC: To Fall Back From a Failed Boot Environment Activation

In this example, the new boot environment was not bootable. You must return to the OK prompt before booting from the original boot environment, c0t4d0s0, in single-user mode.
OK boot net -s
# /sbin/luactivate first_disk
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# init 6

The original boot environment, c0t4d0s0, becomes the active boot environment.

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

In this example, the new boot environment was not bootable. You cannot boot from the original boot environment and must use media or a net installation image. The device is /dev/dsk/c0t4d0s0. The original boot environment, c0t4d0s0, becomes the active boot environment.

OK boot net -s
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
# 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.
  
  
  ```bash
  ./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:

   ```bash
   # /usr/sbin/lu
   ```
2. From the main menu, select Create.

   Name of Current Boot Environment: c0t4d0s0
   Name of New Boot Environment: c0t15d0s0

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 407
 CHAPTER 36

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate an inactive boot environment.</td>
<td>luactivate(1M)</td>
</tr>
<tr>
<td>Cancel a scheduled copy or create job.</td>
<td>lucancel(1M)</td>
</tr>
<tr>
<td>Compare an active boot environment with an inactive boot environment.</td>
<td>lucompare(1M)</td>
</tr>
<tr>
<td>Recopy file systems to update an inactive boot environment.</td>
<td>lucopy</td>
</tr>
<tr>
<td>Create a boot environment.</td>
<td>lucreate(1M)</td>
</tr>
<tr>
<td>Name the active boot environment.</td>
<td>lucurr(1M)</td>
</tr>
<tr>
<td>Delete a boot environment.</td>
<td>ludelete(1M)</td>
</tr>
<tr>
<td>Add a description to a boot environment name.</td>
<td>ludec(1M)</td>
</tr>
<tr>
<td>List critical file systems for each boot environment.</td>
<td>lufslist(1M)</td>
</tr>
<tr>
<td>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.</td>
<td>lumount(1M)</td>
</tr>
<tr>
<td>Rename a boot environment.</td>
<td>lurename(1M)</td>
</tr>
<tr>
<td>List status of all boot environments.</td>
<td>lustatus(1M)</td>
</tr>
<tr>
<td>Task</td>
<td>Command</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>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.</td>
<td>luumount(1M)</td>
</tr>
<tr>
<td>Upgrade an operating environment or install a flash archive on an inactive boot environment.</td>
<td>luupgrade(1M)</td>
</tr>
</tbody>
</table>
Solaris Software Reference (Topics)

This section provides reference information about the Solaris CDs, locales, and packages.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 38</td>
<td>Describes the primary DVD and CDs that are included in the media kits for Solaris 9.</td>
</tr>
<tr>
<td>Chapter 39</td>
<td>Contains a list of the platform names and groups of various hardware platforms.</td>
</tr>
<tr>
<td>Chapter 40</td>
<td>Contains a list of the values that are needed to set the <code>locale</code> keyword in a profile or to preconfigure a locale.</td>
</tr>
</tbody>
</table>
Organization of Solaris 9 Media
(Reference)

This chapter describes the primary DVD and CD media that are included in the media kits for Solaris 9 software.

SPARC: Solaris 9 Media

The following tables list the primary DVD and CD media for the Solaris 9 release. You receive a multilingual media kit that contains DVD or CD media for English, other languages, and locale software.

**TABLE 38-1 SPARC: Multilingual DVD Media**

<table>
<thead>
<tr>
<th>DVD Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 9 <em>SPARC Platform Edition</em> DVD</td>
<td>Contains the following:</td>
</tr>
<tr>
<td></td>
<td>▪ Software, tools, and configuration information to install the Solaris product and all partial locales</td>
</tr>
<tr>
<td></td>
<td>▪ ExtraValue software that includes supported and unsupported software</td>
</tr>
<tr>
<td></td>
<td>▪ Installers</td>
</tr>
<tr>
<td></td>
<td>▪ Source code for some third-party public domain software</td>
</tr>
<tr>
<td></td>
<td>▪ Interface software and documentation that have been localized</td>
</tr>
<tr>
<td></td>
<td>▪ The Solaris documentation set for English, European, and Asian languages that includes Japanese</td>
</tr>
<tr>
<td></td>
<td>For directory structures, see Figure 38-1</td>
</tr>
</tbody>
</table>
### TABLE 38-2 SPARC: Multilingual CD Media

<table>
<thead>
<tr>
<th>CD Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 9 Multilingual Installation</td>
<td>Contains scripts and localized installers to install Solaris software and all partial locales. For directory structures, see Figure 38–2.</td>
</tr>
<tr>
<td>SPARC Platform Edition CD</td>
<td></td>
</tr>
<tr>
<td>Solaris 9 Software 1 of 2 SPARC</td>
<td>Contains the software, tools, and configuration information to install the Solaris product. For directory structures, see Figure 38–3.</td>
</tr>
<tr>
<td>Platform Edition CD</td>
<td></td>
</tr>
<tr>
<td>Solaris 9 Software 2 of 2 SPARC</td>
<td>Contains the following:</td>
</tr>
<tr>
<td>Platform Edition CD</td>
<td><strong>- A limited number of packages, which the software prompts you to install if necessary</strong></td>
</tr>
<tr>
<td></td>
<td><strong>- ExtraValue software that includes supported and unsupported software</strong></td>
</tr>
<tr>
<td></td>
<td><strong>- Installers</strong></td>
</tr>
<tr>
<td></td>
<td><strong>- Source code for some third-party public domain software</strong></td>
</tr>
<tr>
<td></td>
<td>For directory structures, see Figure 38–4.</td>
</tr>
<tr>
<td></td>
<td>The installation program prompts you for this CD if necessary.</td>
</tr>
<tr>
<td>Solaris 9 SPARC Platform Edition Languages CD</td>
<td>Contains interface software and documentation that has been localized. For directory structures, see Figure 38–5.</td>
</tr>
<tr>
<td></td>
<td>The installation program prompts you for this CD if necessary to support languages for specific geographic regions.</td>
</tr>
<tr>
<td>Solaris 9 Documentation 1 of 2 CD</td>
<td>Contains the Solaris documentation set for English and European languages.</td>
</tr>
<tr>
<td>Solaris 9 Documentation 2 of 2 CD</td>
<td>Contains the Solaris documentation set for Asian languages that includes Japanese.</td>
</tr>
</tbody>
</table>

---

**SPARC: Directory Organization of Solaris 9 Media**

This section describes top-level directories on each DVD and CD.

The following figure shows the directory structure on the Solaris 9 SPARC Platform Edition DVD.

![Diagram of Solaris 9 SPARC Platform Edition DVD directory structure]

- **Slice 0 (s0)** contains the `Copyright` and `Solaris_9` directories. The `Solaris_9` directory contains all the tools, software, and configuration information necessary to install the Solaris 9 software product. The `Solaris_9` directory contains the following directories:
  - **ExtraValue** – Two subdirectories:
    - **CoBundled** – Fully supported Solaris products not directly part of the Solaris operating environment that includes Solaris Web Start Wizards SDK 3.0.1 and SunScreen 3.2
    - **EarlyAccess** – Preliminary evaluation software
  - **Misc** – The `jumpstart_sample` directory, which includes a rules file, a check script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
  - **Patches** – All the Solaris 9 patches available at the time of this release.
  - **Product** – The Solaris 9 packages and control files.
  - **Tools** – The Solaris 9 installation tools, which include the following:
    - The **Boot** subdirectory, which contains the Solaris 9 mininroot.
    - The scripts `add_install_client`, `dial`, `rm_install_client`, and `setup_install_server`.

---

Chapter 38 • Organization of Solaris 9 Media (Reference) 419
An Installers subdirectory that contains installers and supporting Java™ class files to install Solaris Live Upgrade and the Solaris 9 Software 2 of 2 SPARC Platform Edition CD. Solaris Live Upgrade is automatically installed when you install the Solaris software. However, in order to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see “To Install Solaris Live Upgrade” on page 339.

- 1of2_Doc_CD – Documentation for English and European Languages
- 2of2_Doc_CD – Documentation for Asian Languages
- Slice 1 (s1) contains the Solaris 9 miniroot.

SPARC: Solaris 9 Installation CD Directory Structure

The following figure shows the directory structure on the Solaris 9 Multilingual Installation SPARC Platform Edition CD or Solaris 9 SPARC Platform Edition Installation CD.

```
+--- multi_icd_sol_9_sparc
    +--- s1
    |     Solaris Boot Image
    +--- s0
        +--- Boot
            +--- Copyright
            +--- add_install_client
            +--- modify_install_server
            +--- rm_install_client
```

FIGURE 38-2 SPARC: Solaris 9 SPARC Platform Edition Installation CD

The directory multi_icd_sol_9_sparc contains scripts that install the Solaris software and includes the Boot subdirectory for the Solaris 9 miniroot. The directories and scripts include the following:

- Slice 0 (s0) contains scripts that install the Solaris software and a Boot subdirectory that contains the Solaris miniroot. These scripts include the following:
  - add_install_client
  - modify_install_server
  - rm_install_client
- Slice 1 (s1) contains the Solaris 9 miniroot.
SPARC: Solaris 9 Software SPARC Platform Edition CDs Directory Structures

The following figures show the directory structure on the Solaris 9 Software SPARC Platform Edition CDs.

![Diagram of directory structure]

**FIGURE 38-3** SPARC: Solaris 9 Software 1 of 2 SPARC Platform Edition CD

- **Slice 0 (s0)** contains the Copyright and Solaris_9 directories. The Solaris_9 directory contains all the tools, software, and configuration information necessary to install, at a minimum, the Solaris 9 software product, including the Core System Support and End User System Support software groups. The Solaris_9 directory contains the following directories:
  - **Docs** - An empty directory.
  - **Misc** - The jumpstart_sample directory, which includes a rules file, a check script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
  - **Patches** - All the Solaris 9 patches available at the time the Solaris 9 Software 1 of 2 SPARC Platform Edition CD was created.
  - **Product** - The Solaris 9 packages and control files.
  - **Tools** - The Solaris 9 installation tools, which include 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 SPARC Platform Edition CD.
The sol_9_sparc_2 directory contains Copyright, Solaris Web Start installer, and Solaris_9 directories. The Solaris_9 directory contains the following subdirectories.

- **ExtraValue** – Two subdirectories:
  - **CoBundled** – Fully supported Solaris products not directly part of the Solaris operating environment
  - **EarlyAccess** – Preliminary evaluation software
- **Product** – A limited number of packages for the Developer System Support, Entire Distribution, and Entire Distribution Plus OEM Support software groups
- **Tools** – Solaris 9 installation tools that include the following:
  - The `add_to_install_server` script
  - An `Installers` subdirectory that contains installers and supporting Java class files to install Solaris Live Upgrade and the Solaris 9 Software 2 of 2 SPARC Platform Edition CD. Solaris Live Upgrade is automatically installed when you install the Solaris software, but to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see “To Install Solaris Live Upgrade” on page 339

**SPARC: Solaris 9 SPARC Platform Edition Languages CD Directory Structure**

The following figure shows the directory structure of the Solaris 9 SPARC Platform Edition 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` contains SimplifiedChinese, French, German, Italian, Japanese, Korean, Spanish, Swedish, TraditionalChinese, and TraditionalChineseHongKong locale packages and packages that are shared by all locales.
- `Copyright` contains the Copyright page.
- `Installer` contains the Solaris Web Start installer.
- `Tools` contains the `add_to_install_server` script for creating a combined installation image.

For more information on locale software, see *International Language Environments Guide*. 
x86: Solaris 9 Media

The following tables list the primary DVD and CD media for Solaris 9 x86 Platform Edition. You receive a multilingual media kit that contains DVD or CD media for English, other languages, and locale software.

### TABLE 38-3 x86: Multilingual DVD Media

<table>
<thead>
<tr>
<th>DVD Title</th>
<th>Description</th>
</tr>
</thead>
</table>
| Solaris 9 x86 Platform Edition DVD | Contains the following:  
- Software, tools, and configuration information to install the Solaris product and all partial locales. This includes the Solaris 9 Device Configuration Assistant, which performs various configuration and booting tasks.  
- ExtraValue software, which includes supported and unsupported software.  
- Installers.  
- Source code for some third-party public domain software.  
- Interface software and documentation that have been localized.  
- The Solaris documentation set for English, European, and Asian languages that includes Japanese.  
For directory structures, see Figure 38–6 |

For directory structures, see Figure 38–6

### TABLE 38–4 x86: Multilingual CD Media

<table>
<thead>
<tr>
<th>CD Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 9 Installation Multilingual x86 Platform Edition CD</td>
<td>Contains scripts and localized installers to install Solaris software and all partial locales. This includes the Solaris 9 Device Configuration Assistant, which performs various configuration and booting tasks. For directory structures, see Figure 38–7.</td>
</tr>
</tbody>
</table>

| Solaris 9 Software 1 of 2 x86 Platform Edition CD | Contains the software, tools, and configuration information to install the Solaris product. This includes the Solaris 9 Device Configuration Assistant, which performs various configuration and booting tasks. For directory structures, see Figure 38–8. |
### Table 38-4: Multilingual CD Media (Continued)

<table>
<thead>
<tr>
<th>CD Title</th>
<th>Description</th>
</tr>
</thead>
</table>
| Solaris 9 Software 2 of 2 x86 Platform Edition CD | Contains the following:  
- A limited number of packages, which the software prompts you to install if necessary  
- ExtraValue software that includes supported and unsupported software  
- Installers  
- Source code for some third-party public domain software  
For directory structures, see Figure 38-9  
The installation program prompts you for this CD if necessary. |
| Solaris 9 x86 Platform Edition Languages CD   | Contains interface software and documentation that are localized. For directory structures, see Figure 38-10  
The installation program prompts you for this CD if necessary to support languages for specific geographic regions. |
| 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.                                                                                                                         |

---

### x86: Directory Organization of Solaris 9 Media

This section describes top-level directories on each DVD and CD.

**Solaris 9 x86 Platform Edition DVD Directory Structure**

The following figure shows the directories on the Solaris 9 x86 Platform Edition DVD.
FIGURE 38–6 Solaris 9 x86 Platform Edition DVD

- Slice 0 (s0) contains the Solaris 9 miniroot.
- Slice 2 (s2) contains the Copyright and Solaris_9 directories. The Solaris_9 directory contains all the tools, software, and configuration information necessary to install the Solaris 9 software product. Slice 2 contains the following directories:
  - **ExtraValue** – Two subdirectories:
    - **CoBundled** – Fully supported Solaris products not directly part of the Solaris operating environment that includes Solaris Web Start Wizards SDK 3.0.1 and SunScreen 3.2
    - **EarlyAccess** – Preliminary evaluation software
  - **Misc** – The jumpstart_sample directory, which includes a rules file, a check script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
  - **Patches** – All the Solaris 9 patches available at the time of this release.
  - **Product** – The Solaris 9 packages and control files.
  - **Tools** – The Solaris 9 installation tools, which include the following.
    - The **Boot** subdirectory that contains the Solaris 9 miniroot.
    - The scripts **add_install_client**, **dial**, **rm_install_client**, and **setup_install_server**
    - An **Installers** subdirectory that contains installers and supporting Java class files to install Solaris Live Upgrade and the Solaris 9 Software 2 of 2 x86 Platform Edition CD. Solaris Live Upgrade is automatically installed when you install the Solaris software. However, in order to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see “To Install Solaris Live Upgrade” on page 339.
x86: Solaris 9 Installation CD Directory Structure

The following figure shows the directories on the Solaris 9 Installation Multilingual x86 Platform Edition CD or Solaris 9 x86 Platform Edition Installation CD.

The directory `multi_icd_sol_9_x86` contains scripts that install the Solaris software and includes the `Boot` subdirectory for the Solaris 9 miniroot. The directories and scripts include the following:

- **Slice 0 (s0)** contains the Solaris 9 miniroot.
- **Slice 2 (s2)** contains scripts that install the Solaris software and a `Boot` subdirectory that contains the Solaris miniroot. These scripts include the following:
  - `add_install_client`
  - `modify_install_server`
  - `rm_install_client`

x86: Solaris 9 Software x86 Platform Edition CDs Directory Structures

The following figures show the directory structure of the Solaris 9 Software x86 Platform Edition CDs.
Slice 0 (s0) contains the Solaris 9 miniroot.

Slice 2 (s2) contains the Copyright and Solaris_9 directories. The Solaris_9 directory contains all the tools, software, and configuration necessary to install, at a minimum, the Solaris 9 software product, including the Core System Support and End User System Support software groups. The Solaris_9 directory contains the following directories:

- **Docs** – Empty directory.
- **Misc** – The jumpstart_sample directory, which includes a rules file, a check script, profiles, begin scripts, finish scripts, and other JumpStart software and files.
- **Patches** – All the Solaris 9 patches available at the time the Solaris 9 Software 1 of 2 x86 Platform Edition CD was created.
- **Product** – The Solaris 9 packages and control files.
- **Tools** – The Solaris 9 installation tools, which include `boot miniroot`, `add_install_client`, `dial`, `rm_install_client`, and `setup_install_server`.

The following figure shows the directories on the Solaris 9 Software 2 of 2 x86 Platform Edition CD.
The `sol_9_x86_2` directory contains `Copyright`, `Solaris Web Start installer`, and `Solaris_9` directories. The `Solaris_9` directory contains the following:

- **ExtraValue**: Two subdirectories:
  - **CoBundled**: Fully supported Solaris products not directly part of the Solaris operating environment
  - **EarlyAccess**: Preliminary evaluation software
- **Product**: A limited number of packages, Developer System Support, Entire Distribution, and Entire Distribution Plus OEM Support software groups.
- **Tools**: This directory includes the following:
  - The `add_to_install_server` script.
  - An `Installer` subdirectory that contains an installer and supporting Java class files to install Solaris Live Upgrade and the Solaris 9 Software 2 of 2 x86 Platform Edition CD. Solaris Live Upgrade is automatically installed when you install the Solaris software. However, in order to use Live Upgrade to upgrade from a previous Solaris release, you need to install these packages first on your current release. For instructions on installing Solaris Live Upgrade, see “To Install Solaris Live Upgrade” on page 339.
  - A `d1_image` subdirectory that contains the Solaris 9 Device Configuration Assistant software that can be copied to a diskette.

### x86: Solaris 9 x86 Platform Edition Languages CD

**Directory Structures**

The following figure shows the directory structure of the Solaris 9 x86 Platform Edition Languages CD.
The `sol_9_lang_x86` directory contains the Solaris Web Start program, which installs the Solaris language and locale software. You can choose to install the nine languages in default directories or install specific languages. The `sol_9_lang_x86` directory contains these directories:

- **Components** – Contains SimplifiedChinese, French, German, Italian, Japanese, Korean, Spanish, Swedish, TraditionalChinese, and TraditionalChineseHongKong locale packages and also includes packages that are shared by all locales.
- **Copyright** – The copyright page
- **Installer** – The Web Start installer
- **Tools** – The `add_to_install_server` script for creating an install server

For more information on locale software, see *International Language Environments Guide*. 
Platform Names and Groups
(Reference)

You need to know your system architecture (platform group) if you are adding clients for a network installation and the platform name if you are writing a custom JumpStart installation rules file.

Some examples of platform names and groups follow. For a full list of SPARC systems, see Solaris 9 Sun Hardware Platform Guide.

### TABLE 39–1 Example of Platform Names and Groups

<table>
<thead>
<tr>
<th>System</th>
<th>Platform Name</th>
<th>Platform Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultra 5</td>
<td>SUNW,Ultra-5_10</td>
<td>sun4u</td>
</tr>
<tr>
<td>x86 based</td>
<td>i86pc</td>
<td>i86pc</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>hi_IN.UTF-8</td>
<td>English</td>
<td>India</td>
<td>UTF-8(^1)</td>
<td>Hindi (UTF-8) Unicode 3.1</td>
</tr>
<tr>
<td>ja</td>
<td>Japanese</td>
<td>Japan</td>
<td>eucJP(^2)</td>
<td>Japanese (EUC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JIS X 0201-1976</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JIS X 0208-1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JIS X 0212-1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JIS X 0201-1976</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JIS X 0208-1990</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JIS X 0212-1990</td>
</tr>
<tr>
<td>ja_JP.PCK</td>
<td>Japanese</td>
<td>Japan</td>
<td>PCK(^3)</td>
<td>Japanese (PC kanji)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JIS X 0201-1976</td>
</tr>
</tbody>
</table>

UTF-8 is the UTF-8 defined in ISO/IEC 10646-1:2000 and also Unicode 3.1.
PCK is also known as Shift_JIS (SJIS).
### TABLE 40–1 Asia (Continued)

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ja_JP.UTF-8</td>
<td>Japanese</td>
<td>Japan</td>
<td>UTF-8</td>
<td>Japanese (UTF-8) Unicode 3.1</td>
</tr>
<tr>
<td>ko_KR.EUC</td>
<td>Korean</td>
<td>Korea</td>
<td>1001</td>
<td>Korean (EUC) KS X 1001</td>
</tr>
<tr>
<td>ko_KR.UTF-8</td>
<td>Korean</td>
<td>Korea</td>
<td>UTF-8</td>
<td>Korean (UTF-8) Unicode 3.1</td>
</tr>
<tr>
<td>th_TH.UTF-8</td>
<td>English</td>
<td>Thailand</td>
<td>UTF-8</td>
<td>Thai (UTF-8) Unicode 3.1</td>
</tr>
<tr>
<td>th_TH.TIS620</td>
<td>English</td>
<td>Thailand</td>
<td>TIS620.2533</td>
<td>Thai TIS620.2533</td>
</tr>
<tr>
<td>zh_CN.EUC</td>
<td>Simplified Chinese</td>
<td>PRC</td>
<td>gb2312⁴</td>
<td>Simplified Chinese (EUC) GB2312-1980</td>
</tr>
<tr>
<td>zh_CN.GBK</td>
<td>Simplified Chinese</td>
<td>PRC</td>
<td>GBK⁵</td>
<td>Simplified Chinese (GBK)</td>
</tr>
<tr>
<td>zh_CN.UTF-8</td>
<td>Simplified Chinese</td>
<td>PRC</td>
<td>UTF-8</td>
<td>Simplified Chinese (UTF-8) Unicode 3.1</td>
</tr>
<tr>
<td>zh_HK.BIG5HK</td>
<td>Traditional Chinese</td>
<td>Hong Kong</td>
<td>Big5+HKSCS</td>
<td>Traditional Chinese (BIG5+HKSCS)</td>
</tr>
<tr>
<td>zh_HK.UTF-8</td>
<td>Traditional Chinese</td>
<td>Hong Kong</td>
<td>UTF-8</td>
<td>Traditional Chinese (UTF-8) Unicode 3.1</td>
</tr>
<tr>
<td>zh_TW.EUC</td>
<td>Traditional Chinese</td>
<td>Taiwan</td>
<td>cns11643</td>
<td>Traditional Chinese (EUC) CNS 11643-1992</td>
</tr>
<tr>
<td>zh_TW.BIG5</td>
<td>Traditional Chinese</td>
<td>Taiwan</td>
<td>BIG5</td>
<td>Traditional Chinese (BIG5)</td>
</tr>
<tr>
<td>zh_TW.UTF-8</td>
<td>Traditional Chinese</td>
<td>Taiwan</td>
<td>UTF-8</td>
<td>Traditional Chinese (UTF-8) Unicode 3.1</td>
</tr>
</tbody>
</table>

GBK signifies GB extensions. This includes all GB 2312–80 characters and all Unified Han characters of ISO/IEC 10646–1, as well as Japanese Hiragana and Katakana characters. It also includes many characters of Chinese, Japanese, and Korean character sets and of ISO/IEC 10646–1.

### TABLE 40–2 Australasia

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>en_AU.ISO8859-1</td>
<td>English</td>
<td>Australia</td>
<td>ISO8859-1</td>
<td>English (Australia)</td>
</tr>
<tr>
<td>en_NZ.ISO8859-1</td>
<td>English</td>
<td>New Zealand</td>
<td>ISO8859-1</td>
<td>English (New Zealand)</td>
</tr>
</tbody>
</table>
### TABLE 40-3 Central America

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>es_CR.ISO8859-1</td>
<td>Spanish</td>
<td>Costa Rica</td>
<td>ISO8859-1</td>
<td>Spanish (Costa Rica)</td>
</tr>
<tr>
<td>es_GT.ISO8859-1</td>
<td>Spanish</td>
<td>Guatemala</td>
<td>ISO8859-1</td>
<td>Spanish (Guatemala)</td>
</tr>
<tr>
<td>es_NI.ISO8859-1</td>
<td>Spanish</td>
<td>Nicaragua</td>
<td>ISO8859-1</td>
<td>Spanish (Nicaragua)</td>
</tr>
<tr>
<td>es_PA.ISO8859-1</td>
<td>Spanish</td>
<td>Panama</td>
<td>ISO8859-1</td>
<td>Spanish (Panama)</td>
</tr>
<tr>
<td>es_SV.ISO8859-1</td>
<td>Spanish</td>
<td>El Salvador</td>
<td>ISO8859-1</td>
<td>Spanish (El Salvador)</td>
</tr>
</tbody>
</table>

### TABLE 40-4 Central Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs_CZ.ISO8859-2</td>
<td>English</td>
<td>Czech Republic</td>
<td>ISO8859-2</td>
<td>Czech (Czech Republic)</td>
</tr>
<tr>
<td>de_AT.ISO8859-1</td>
<td>German</td>
<td>Austria</td>
<td>ISO8859-1</td>
<td>German (Austria)</td>
</tr>
<tr>
<td>de_AT.ISO8859-15</td>
<td>German</td>
<td>Austria</td>
<td>ISO8859-15</td>
<td>German (Austria, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>de_CH.ISO8859-1</td>
<td>German</td>
<td>Switzerland</td>
<td>ISO8859-1</td>
<td>German (Switzerland)</td>
</tr>
<tr>
<td>de_DE.UTF-8</td>
<td>German</td>
<td>Germany</td>
<td>UTF-8</td>
<td>German (Germany, Unicode 3.1)</td>
</tr>
<tr>
<td>de_DE.ISO8859-1</td>
<td>German</td>
<td>Germany</td>
<td>ISO8859-1</td>
<td>German (Germany)</td>
</tr>
<tr>
<td>de_DE.ISO8859-15</td>
<td>German</td>
<td>Germany</td>
<td>ISO8859-15</td>
<td>German (Germany, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>fr_CH.ISO8859-1</td>
<td>French</td>
<td>Switzerland</td>
<td>ISO8859-1</td>
<td>French (Switzerland)</td>
</tr>
<tr>
<td>hu_HU.ISO8859-2</td>
<td>English</td>
<td>Hungary</td>
<td>ISO8859-2</td>
<td>Hungarian (Hungary)</td>
</tr>
<tr>
<td>pl_PL.ISO8859-2</td>
<td>English</td>
<td>Poland</td>
<td>ISO8859-2</td>
<td>Polish (Poland)</td>
</tr>
<tr>
<td>pl_PL.UTF-8</td>
<td>English</td>
<td>Poland</td>
<td>UTF-8</td>
<td>Polish (Poland, Unicode 3.1)</td>
</tr>
<tr>
<td>sk_SK.ISO8859-2</td>
<td>English</td>
<td>Slovakia</td>
<td>ISO8859-2</td>
<td>Slovak (Slovakia)</td>
</tr>
</tbody>
</table>

### TABLE 40-5 Eastern Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>bg_BG.ISO8859-5</td>
<td>English</td>
<td>Bulgaria</td>
<td>ISO8859-5</td>
<td>Bulgarian (Bulgaria)</td>
</tr>
<tr>
<td>et_EE.ISO8859-15</td>
<td>English</td>
<td>Estonia</td>
<td>ISO8859-15</td>
<td>Estonian (Estonia)</td>
</tr>
</tbody>
</table>
TABLE 40-5 Eastern Europe (Continued)

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>hr_HR.ISO8859-2</td>
<td>English</td>
<td>Croatia</td>
<td>ISO8859-2</td>
<td>Croatian (Croatia)</td>
</tr>
<tr>
<td>lt_LT.ISO8859-13</td>
<td>English</td>
<td>Lithuania</td>
<td>ISO8859-13</td>
<td>Lithuanian (Lithuania)</td>
</tr>
<tr>
<td>lv_LV.ISO8859-13</td>
<td>English</td>
<td>Latvia</td>
<td>ISO8859-13</td>
<td>Latvian (Latvia)</td>
</tr>
<tr>
<td>mk_MK.ISO8859-5</td>
<td>English</td>
<td>Macedonia</td>
<td>ISO8859-5</td>
<td>Macedonian (Macedonia)</td>
</tr>
<tr>
<td>ro_RO.ISO8859-2</td>
<td>English</td>
<td>Romania</td>
<td>ISO8859-2</td>
<td>Romanian (Romania)</td>
</tr>
<tr>
<td>ru_RU.KOI8-R</td>
<td>English</td>
<td>Russia</td>
<td>Koi8-R</td>
<td>Russian (Russia, KOI8-R)</td>
</tr>
<tr>
<td>ru_RU.ANSI1251</td>
<td>English</td>
<td>Russia</td>
<td>ansi-1251</td>
<td>Russian (Russia, ANSI 1251)</td>
</tr>
<tr>
<td>ru_RU.ISO8859-5</td>
<td>English</td>
<td>Russia</td>
<td>ISO8859-5</td>
<td>Russian (Russia)</td>
</tr>
<tr>
<td>ru_RU.UTF-8</td>
<td>English</td>
<td>Russia</td>
<td>UTF-8</td>
<td>Russian (Russia, Unicode 3.1)</td>
</tr>
<tr>
<td>sh_BA.ISO8859-2@bosnia</td>
<td>English</td>
<td>Bosnia</td>
<td>ISO8859-2</td>
<td>Bosnian (Bosnia)</td>
</tr>
<tr>
<td>sl_SI.ISO8859-2</td>
<td>English</td>
<td>Slovenia</td>
<td>ISO8859-2</td>
<td>Slovenian (Slovenia)</td>
</tr>
<tr>
<td>sq_AL.ISO8859-2</td>
<td>English</td>
<td>Albania</td>
<td>ISO8859-2</td>
<td>Albanian (Albania)</td>
</tr>
<tr>
<td>sr_YU.ISO8859-5</td>
<td>English</td>
<td>Serbia</td>
<td>ISO8859-5</td>
<td>Serbian (Serbia)</td>
</tr>
<tr>
<td>tr_TR.ISO8859-9</td>
<td>English</td>
<td>Turkey</td>
<td>ISO8859-9</td>
<td>Turkish (Turkey)</td>
</tr>
<tr>
<td>tr_TR.UTF-8</td>
<td>English</td>
<td>Turkey</td>
<td>UTF-8</td>
<td>Turkish (Turkey, Unicode 3.1)</td>
</tr>
</tbody>
</table>

TABLE 40-6 Middle East

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>He</td>
<td>English</td>
<td>Israel</td>
<td>ISO8859-8</td>
<td>Hebrew (Israel)</td>
</tr>
</tbody>
</table>

TABLE 40-7 North Africa

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar_EG.UTF-8</td>
<td>English</td>
<td>Egypt</td>
<td>UTF-8</td>
<td>Arabic (Egypt)</td>
</tr>
</tbody>
</table>
### TABLE 40–7 North Africa (Continued)

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ar</td>
<td>English</td>
<td>Egypt</td>
<td>ISO8859-6</td>
<td>Arabic (Egypt)</td>
</tr>
</tbody>
</table>

### TABLE 40–8 North America

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>en_CA.ISO8859-1</td>
<td>English</td>
<td>Canada</td>
<td>ISO8859-1</td>
<td>English (Canada)</td>
</tr>
<tr>
<td>en_US.ISO8859-1</td>
<td>English</td>
<td>USA</td>
<td>ISO8859-1</td>
<td>English (U.S.A.)</td>
</tr>
<tr>
<td>en_US.UTF-8</td>
<td>English</td>
<td>USA</td>
<td>UTF-8</td>
<td>English (U.S.A., Unicode 3.1)</td>
</tr>
<tr>
<td>fr_CA.ISO8859-1</td>
<td>French</td>
<td>Canada</td>
<td>ISO8859-1</td>
<td>French (Canada)</td>
</tr>
<tr>
<td>es_MX.ISO8859-1</td>
<td>Spanish</td>
<td>Mexico</td>
<td>ISO8859-1</td>
<td>Spanish (Mexico)</td>
</tr>
</tbody>
</table>

### TABLE 40–9 Northern Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>da_DK.ISO8859-1</td>
<td>English</td>
<td>Denmark</td>
<td>ISO8859-1</td>
<td>Danish (Denmark)</td>
</tr>
<tr>
<td>da_DK.ISO8859-15</td>
<td>English</td>
<td>Denmark</td>
<td>ISO8859-15</td>
<td>Danish (Denmark, ISO8859-15 – Euro)</td>
</tr>
<tr>
<td>fi_FI.ISO8859-1</td>
<td>English</td>
<td>Finland</td>
<td>ISO8859-1</td>
<td>Finnish (Unicode 3.1)</td>
</tr>
<tr>
<td>fi_FI.ISO8859-15</td>
<td>English</td>
<td>Finland</td>
<td>ISO8859-15</td>
<td>Finnish (Finland, ISO8859-15–Euro)</td>
</tr>
<tr>
<td>fi_FI.UTF-8</td>
<td>English</td>
<td>Finland</td>
<td>UTF-8</td>
<td>Finnish (Finland)</td>
</tr>
<tr>
<td>is_IS.ISO8859-1</td>
<td>English</td>
<td>Iceland</td>
<td>ISO8859-1</td>
<td>Icelandic (Iceland)</td>
</tr>
<tr>
<td>no_NO.ISO8859-1@bokmal</td>
<td>English</td>
<td>Norway</td>
<td>ISO8859-1</td>
<td>Norwegian (Norway-Bokmal)</td>
</tr>
<tr>
<td>no_NO.ISO8859-1@nyorskr</td>
<td>English</td>
<td>Norway</td>
<td>ISO8859-1</td>
<td>Norwegian (Norway-Nynorsk)</td>
</tr>
<tr>
<td>sv_SE.ISO8859-1</td>
<td>Swedish</td>
<td>Sweden</td>
<td>ISO8859-1</td>
<td>Swedish (Sweden)</td>
</tr>
<tr>
<td>sv_SE.UTF-8</td>
<td>Swedish</td>
<td>Sweden</td>
<td>UTF-8</td>
<td>Swedish (Sweden, Unicode 3.1)</td>
</tr>
</tbody>
</table>
### TABLE 40–10 South America

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>es_AR.ISO8859-1</td>
<td>Spanish</td>
<td>Argentina</td>
<td>ISO8859-1</td>
<td>Spanish (Argentina)</td>
</tr>
<tr>
<td>es_BO.ISO8859-1</td>
<td>Spanish</td>
<td>Bolivia</td>
<td>ISO8859-1</td>
<td>Spanish (Bolivia)</td>
</tr>
<tr>
<td>es_CL.ISO8859-1</td>
<td>Spanish</td>
<td>Chile</td>
<td>ISO8859-1</td>
<td>Spanish (Chile)</td>
</tr>
<tr>
<td>es_CO.ISO8859-1</td>
<td>Spanish</td>
<td>Colombia</td>
<td>ISO8859-1</td>
<td>Spanish (Colombia)</td>
</tr>
<tr>
<td>es_EC.ISO8859-1</td>
<td>Spanish</td>
<td>Ecuador</td>
<td>ISO8859-1</td>
<td>Spanish (Ecuador)</td>
</tr>
<tr>
<td>es_PE.ISO8859-1</td>
<td>Spanish</td>
<td>Peru</td>
<td>ISO8859-1</td>
<td>Spanish (Peru)</td>
</tr>
<tr>
<td>es_PY.ISO8859-1</td>
<td>Spanish</td>
<td>Paraguay</td>
<td>ISO8859-1</td>
<td>Spanish (Paraguay)</td>
</tr>
<tr>
<td>es_UY.ISO8859-1</td>
<td>Spanish</td>
<td>Uruguay</td>
<td>ISO8859-1</td>
<td>Spanish (Uruguay)</td>
</tr>
<tr>
<td>es_VE.ISO8859-1</td>
<td>Spanish</td>
<td>Venezuela</td>
<td>ISO8859-1</td>
<td>Spanish (Venezuela)</td>
</tr>
<tr>
<td>pt_BR.ISO8859-1</td>
<td>English</td>
<td>Brazil</td>
<td>ISO8859-1</td>
<td>Portuguese (Brazil)</td>
</tr>
<tr>
<td>pt_BR.UTF-8</td>
<td>English</td>
<td>Brazil</td>
<td>UTF-8</td>
<td>Portuguese (Brazil, Unicode 3.1)</td>
</tr>
</tbody>
</table>

### TABLE 40–11 South Europe

<table>
<thead>
<tr>
<th>Locale</th>
<th>User Interface</th>
<th>Territory</th>
<th>Codeset</th>
<th>Language Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca_ES.ISO8859-1</td>
<td>English</td>
<td>Spain</td>
<td>ISO8859-1</td>
<td>Catalan (Spain)</td>
</tr>
<tr>
<td>ca_ES.ISO8859-15</td>
<td>English</td>
<td>Spain</td>
<td>ISO8859-15</td>
<td>Catalan (Spain, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>el_GR.ISO8859-7</td>
<td>English</td>
<td>Greece</td>
<td>ISO8859-7</td>
<td>Greek (Greece)</td>
</tr>
<tr>
<td>es_ES.ISO8859-1</td>
<td>Spanish</td>
<td>Spain</td>
<td>ISO8859-1</td>
<td>Spanish (Spain)</td>
</tr>
<tr>
<td>es_ES.ISO8859-15</td>
<td>Spanish</td>
<td>Spain</td>
<td>ISO8859-15</td>
<td>Spanish (Spain, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>es_ES.UTF-8</td>
<td>Spanish</td>
<td>Spain</td>
<td>UTF-8</td>
<td>Spanish (Spain, Unicode 3.1)</td>
</tr>
<tr>
<td>it_IT.ISO8859-1</td>
<td>Italian</td>
<td>Italy</td>
<td>ISO8859-1</td>
<td>Italian (Italy)</td>
</tr>
<tr>
<td>it_IT.ISO8859-15</td>
<td>Italian</td>
<td>Italy</td>
<td>ISO8859-15</td>
<td>Italian (Italy, ISO8859-15 - Euro)</td>
</tr>
<tr>
<td>it_IT.UTF-8</td>
<td>Italian</td>
<td>Italy</td>
<td>UTF-8</td>
<td>Italian (Italy, Unicode 3.1)</td>
</tr>
<tr>
<td>pt_PT.ISO8859-1</td>
<td>English</td>
<td>Portugal</td>
<td>ISO8859-1</td>
<td>Portuguese (Portugal)</td>
</tr>
<tr>
<td>Locale</td>
<td>User Interface</td>
<td>Territory</td>
<td>Codeset</td>
<td>Language Support</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>-----------------</td>
<td>-------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>en_GB.ISO8859-1</td>
<td>English</td>
<td>Great Britain</td>
<td>ISO8859-1</td>
<td>English (Great Britain)</td>
</tr>
<tr>
<td>en_IE.ISO8859-1</td>
<td>English</td>
<td>Ireland</td>
<td>ISO8859-1</td>
<td>English (Ireland)</td>
</tr>
<tr>
<td>fr_BE.ISO8859-1</td>
<td>French</td>
<td>Belgium-Walloon</td>
<td>ISO8859-1</td>
<td>French (Belgium-Walloon, Unicode 3.1)</td>
</tr>
<tr>
<td>fr_BE.UTF-8</td>
<td>French</td>
<td>Belgium-Walloon</td>
<td>UTF-8</td>
<td>French (Belgium-Walloon, Unicode 3.1)</td>
</tr>
<tr>
<td>fr_FR.ISO8859-1</td>
<td>French</td>
<td>France</td>
<td>ISO8859-1</td>
<td>French (France)</td>
</tr>
<tr>
<td>fr_FR.UTF-8</td>
<td>French</td>
<td>France</td>
<td>UTF-8</td>
<td>French (France, Unicode 3.1)</td>
</tr>
<tr>
<td>nl_BE.ISO8859-1</td>
<td>English</td>
<td>Belgium-Flemish</td>
<td>ISO8859-1</td>
<td>Dutch (Belgium-Flemish)</td>
</tr>
<tr>
<td>nl_NL.ISO8859-1</td>
<td>English</td>
<td>Netherlands</td>
<td>ISO8859-1</td>
<td>Dutch (Netherlands)</td>
</tr>
</tbody>
</table>
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 441
- “Problems With Booting a System” on page 442
- “Initial Installation of the Solaris 9 Operating Environment” on page 449
- “Upgrading the Solaris 9 Operating Environment” on page 451

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 (SPARC based systems only)
  **Cause:** This error occurs when you override the location of the boot - file by explicitly setting it to /kernel/unix.

  **Solution:**
  - Reset the boot - file in the PROM to “” (blank).
  - Ensure that the diag-switch is set to off and to true.

Can’t boot from file/device
  **Cause:** The installation media cannot find the bootable media.

  **Solution:** Ensure that the following conditions are met:
  - The DVD-ROM or CD-ROM drive is installed properly and turned on.
  - Solaris 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! (SPARC based systems only)
  **Description:** This is an informational message.

  **Solution:** Ignore the message and continue with the installation.
Not a UFS file system (x86 based systems only)

**Cause:** When Solaris 9 software was installed (either through the Solaris suninstall program or custom JumpStart), no boot disk was selected. You now must use the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette or edit the BIOS to boot the system.

**Solution:**

- Insert the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette into the system’s boot diskette drive (usually drive A). For information on accessing the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette, see “x86: Accessing the Solaris 9 Device Configuration Assistant and PXE” on page 31.
- If you cannot use the bootable media, go into the BIOS and select the BIOS to boot. See your BIOS documentation for instructions.

The Solaris Installer could not find a disk that meets the criteria found in the Install documentation. Please see the documentation for more info. (x86 based systems only)

**Cause:** You’ve tried to boot from the Solaris 9 x86 Platform Edition Installation CD. The system does not support logical block addressing (LBA) and the Solaris 9 Installation CD cannot be used.

**Solution:** Use a net image of the CD, a DVD, or the Solaris 9 Software 1 of 2 x86 Platform Edition CD to install.

---

**Booting From Media, General Problems**

The system does not boot.

**Description:** When initially setting up a custom JumpStart server, you might encounter boot problems that do not return an error message. To verify information about the system and how the system is booting, run the boot command with the `-v` option. When you use the `-v` option, the boot command displays verbose debugging information on the screen.

**Note** – If this flag is not given, the messages are still printed, but the output is directed to the system log file. For more information, see syslogd(1M).

**Solution:** For SPARC based systems, at the `ok` prompt, type the following command.

```
ok boot net -v - install
```

For x86 based systems, when the installation program prompts you to “Select type of installation,” type the following command.
b -v install

Boot from DVD media fails on systems with Toshiba SD-M 1401 DVD-ROM

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

The installation fails after booting. (x86 based systems only)

Cause: If you are installing from the Solaris 9 Installation CD, the Solaris 9 root slice must be located within the first 1024 cylinders of the disk.

Solution: The BIOS and SCSI driver for the default boot disk must support logical block addressing (LBA). LBA enables the machine to boot beyond the 1024-cylinder limit and across Solaris disk slices. To determine if your system supports LBA, see Table 2–4. If your system does not support LBA, boot from a net image rather than the CD.

The system hangs or panics when non-memory PC cards are inserted. (x86 based systems only)

Cause: Non-memory PC cards cannot use the same memory resources used by other devices.

Solution: To correct this problem, see the instructions for your PC card and check for the address range.

The IDE BIOS primary drive on your system was not detected by the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette during the pre-booting phase. (x86 based systems only)

Solution:

- If you are using old drives, they might be unsupported. Check your hardware manufacturer’s documentation.
- Make sure the ribbon and power cables are plugged in correctly. Check the manufacturer’s documentation.
- If only one drive is attached to the controller, designate the drive as the master drive by setting jumpers. Some drives have different jumper settings for a single master, as opposed to a master operating with a slave. Connect the drive to the connector at the end of the cable to reduce signal ringing that occurs when an unused connector is dangling at the end of the cable.
- If two drives are attached to the controller, jumper one drive as the master (or as a master operating with a slave), and jumper the second drive as a slave.
- If one drive is a hard disk and the second a CD-ROM drive, designate one drive as the slave drive by setting jumpers. It does not matter which drive is connected to which drive connection on the cable.
- If problems persist with two drives on a single controller, attach one drive at a time to verify that each drive works. Jumper the drive as master or single master, and use the drive connector at the end of the IDE ribbon cable to attach the drive. Verify that each drive works, then jumper the drives back to a master and slave configuration.
- If the drive is a disk drive, use the BIOS setup utility to ensure that the drive type (which indicates the number of cylinders, heads, and sectors) is configured correctly. Some BIOS software might have a feature that automatically detects the drive type.
- If the drive is a CD-ROM drive, use the BIOS setup screen to configure the drive type as a CD-ROM drive, provided the BIOS software offers this capability.
- For many systems, IDE CD-ROM drives are only recognized by MS-DOS if an MS-DOS CD-ROM driver has been installed. Try another drive.

The IDE disk or CD-ROM drive on your system was not found by the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette during the pre-booting phase. (x86 based systems only)

Solution:

- If disks are disabled in the BIOS, use the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette to boot from the hard disk. For information on accessing the Solaris 9 Device Configuration Assistant, see “x86: Accessing the Solaris 9 Device Configuration Assistant and PXE” on page 31.
- If the system has no disks, it might be a diskless client.

The system hangs before displaying the system prompt. (x86 based systems only)

Solution: You have hardware that is not supported. Check your hardware manufacturer’s documentation.

Booting From the Network, Error Messages


Description: This error occurs when you have two or more servers on a network responding to an install client’s boot request. The install client connects to the wrong boot server, and the installation hangs. The following specific reasons might cause this error to occur:

Cause: Reason 1: /etc/bootparams files might exist on different servers with an entry for this install client.
Solution: Reason 1: Ensure that servers on the network do not have multiple /etc/bootparams entries for the install client. If they do have multiple entries, remove duplicate client entries in the /etc/bootparams file on all install servers and boot servers except the one you want the install client to use.

Cause: Reason 2: Multiple /tftpboot or /rplboot directory entries might exist for this install client.

Solution: Reason 2: Ensure that servers on the network do not have multiple /tftpboot or /rplboot directory entries for the install client. If they do have multiple entries, remove duplicate client entries from the /tftpboot or /rplboot directories on all install servers and boot servers except the one you want the install client to use.

Cause: Reason 3: An install client entry might exist in the /etc/bootparams file on a server and an entry in another /etc/bootparams file that enables all systems to access the profile server. Such an entry resembles the following:

```
* install_config=profile_server:path
```

A line that resembles the previous entry in the NIS or NIS+ bootparams table can also cause this error.

Solution: Reason 3: If a wildcard entry is in the name service bootparams map or table (for example, * install_config=), delete it and add it to the /etc/bootparams file on the boot server.

No network boot server. Unable to install the system. See installation instructions. (SPARC based systems only)

Cause: This error occurs on a system that you are attempting to install from the network. The system is not set up correctly.

Solution: Ensure that you correctly set up the system to install from the network. See “Adding Systems to Be Installed From the Network With a CD Image” on page 119.

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

Cause: This error occurs when you are installing Solaris 9 from a network, but the boot software cannot locate the following:

- Solaris 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... (SPARC based systems only)

Cause: Reason 1: The client is trying to boot from the network, but it cannot find a system that knows about the client.

Solution: Reason 1: Verify the system’s host name is in the NIS or NIS+ name service. Also, verify the bootparams search order in the boot server’s /etc/nsswitch.conf file.

For example, the following line in the /etc/nsswitch.conf file indicates that JumpStart or the Solaris suninstall program first looks in the NIS maps for bootparams information. If the program does not find any information, the JumpStart program or the Solaris suninstall program looks in the boot server’s /etc/bootparams file.

bootparams: nis files

Cause: Reason 2: The client’s Ethernet address is not correct.

Solution: Reason 2: Verify that the client’s Ethernet address in the install server’s /etc/ethers file is correct.

Cause: Reason 3: In a custom JumpStart installation, the add_install_client command specifies the platform group that uses a specified server as an install server. If the wrong architecture value is used when using the add_install_client, this problem occurs. For example, the machine you want to install is a sun4u, but you used sun4m instead.

Solution: Reason 3: Rerun add_install_client with the correct architecture value.

ip: joining multicasts failed on tr0 - will use link layer broadcasts for multicast (x86 based systems only)

Cause: This error message is displayed when you boot a system with a token ring card. Ethernet multicast and token ring multicast do not work the same way. The driver returns this error message because an invalid multicast address was provided to it.

Solution: Ignore this error message. If multicast does not work, IP uses layer broadcasts instead and does not cause the installation to fail.

Requesting Internet address for Ethernet_Address (x86 based systems only)

Cause: The client is trying to boot from the network, but it cannot find a system that knows about the client.
Solution: Verify the system’s host name is listed in the name service. If the system’s host name is listed in the NIS or NIS+ name service, and the system continues to print this error message, try rebooting.

RPC: Timed out No bootparams (whoami) server responding; still trying... (x86 based systems only)
Cause: The client is trying to boot from the network, but it cannot find a system with an entry in the /etc/bootparams file on the install server.

Solution: Use add_install_client on the install server. Using this command adds the proper entry in the /etc/bootparams file, enabling the client to boot from the network.

Still trying to find a RPL server... (x86 based systems only)
Cause: The system is trying to boot from the network, but the server is not set up to boot this system.

Solution: On the install server, execute add_install_client for the system to be installed. The add_install_client command sets up an /rplboot directory, which contains the necessary network boot program.

Booting From the Network, General Problems

The system boots from the network, but from a system other than the specified install server.

Cause: An /etc/bootparams and perhaps /etc/ethers entry exist on another system for the client.

Solution: On the name server, update the /etc/bootparams entry for the system that is being installed. The entry should conform to the following syntax:

```
install_system root=boot_server:path install=install_server:path
```

Also, ensure that only one bootparams entry is on the subnet for the install client.

After you set up an install server and configure the system to install Solaris 9 from the network, the system still does not boot. (SPARC based systems only)

Cause: The tftpd might not be running on the install server.

Solution: Be sure the tftpd daemon is running on the install server. Type the following command:

```
# ps -ef | grep tftpd
```

If this command does not return a line that indicates that the tftpd daemon is running, edit the /etc/inetd.conf file and remove the comment (＃) character from the following line:
After setting up an install server and configuring the system to install from the network, the system still does not boot. (*x86 based systems only*)

**Cause:** The rpld daemon might not be running on the install server.

**Solution:** Be sure the rpld daemon is running on the install server. Type the following command:

```
# ps -ef | grep rpld
```

If this command does not return a line indicating the rpld daemon is running, execute the following command:

```
# /usr/sbin/rpld
```

After making this change, try booting the system again.

---

**Initial Installation of the Solaris 9 Operating Environment**

**Initial installation fails**

**Solution:** If the Solaris installation fails, you must restart the installation. To restart the installation, boot the system from the Solaris 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/SUNWxxxx/reloc.cpio: Broken pipe
```

**Description:** This error message does not affect the installation.

**Solution:** Ignore the message and continue with the installation.

**WARNING: CHANGE DEFAULT BOOT DEVICE (*x86 based systems only*)**

**Cause:** This is an informational message. The default boot device set in the system’s BIOS might be set to a device that requires you to use the Solaris 9 Device Configuration Assistant *x86 Platform Edition* diskette to boot the system.
Solution: Continue with the installation and, if necessary, change the system’s default boot device specified in the BIOS after you install the Solaris software to a device that does not require the Solaris 9 Device Configuration Assistant x86 Platform Edition diskette.

▼ x86: To Check IDE Disk for Bad Blocks

IDE disk drives do not automatically map out bad blocks like other drives supported by Solaris software. Before installing Solaris 9 on an IDE disk, you might want to perform a surface analysis on the disk. To perform surface analysis on an IDE disk, follow this procedure.

1. Boot to the installation media in single-user mode.
   
   ```
   # b -s
   ```

2. Start the format program.
   
   ```
   # format
   ```

3. Specify the IDE disk drive on which you want to perform a surface analysis.
   
   ```
   # cxdy
   ```
   
   cx  Is the controller number
   dy  Is the device number

4. You need an fdisk partition.
   
   - If a Solaris fdisk partition already exists, proceed to Step 5
   - If a Solaris fdisk partition does not exist, use the fdisk command to create a Solaris partition on the disk.

   ```
   format> fdisk
   ```

5. Type:
   
   ```
   format> analyze
   ```

6. Type:
   
   ```
   analyze> config
   ```
   The current settings for a surface analysis are displayed.

   a. If you want to change settings, type:
      
      ```
      analyze> setup
      ```

7. Type:
analyze> type_of_surface_analysis

_type_of_surface_analysis_ Is read, write, or compare

If format finds bad blocks, it re-maps them.

8. Type:

   analyze> quit

9. Do you want to specify blocks to re-map?
   - If no, go to Step 10
   - If yes, type:
     
     format> repair

10. Type:

    quit
    The format program quits.

11. To restart the media in multiuser mode, type:

    ok b

---

Upgrading the Solaris 9 Operating Environment

Upgrading, Error Messages

**No upgradable disks**

*Cause:* A swap entry in the `/etc/vfstab` file is causing the upgrade to fail.

*Solution:* Comment out the following lines in the `/etc/vfstab` file:

- All swap files and slices on disks not being upgraded
- Swap files that are no longer present
- Any unused swap slices

**usr/bin/bzczt not found**

*Cause:* Solaris Live Upgrade fails because of needing a patch cluster.
Solution: A patch is needed to install Solaris Live Upgrade. Go to http://sunsolve.sun.com for the patch.

Upgradeable Solaris root devices were found, however, no suitable partitions to hold the Solaris install software were found. Upgrading using the Solaris Installer is not possible. It might be possible to upgrade using the Solaris Operating Environment 1 of 2 CDROM. (x86 based systems only)

Cause: You cannot upgrade with Solaris 9 x86 Platform Edition Installation CD because you do not have enough space.

Solution: To upgrade, you can either create a swap slice that is larger than or equal to 512 Mbytes or use another method of upgrading such as the following:

- The Solaris Web Start program from Solaris 9 DVD or a net installation image
- The Solaris suninstall program from the Solaris 9 Software 1 of 2 x86 Platform Edition CD
- Custom JumpStart

Upgrading, General Problems

The upgrade option is not presented even though there is a version of Solaris software that’s upgradable on the system.

Cause: Reason 1: The /var/sadm directory is a symlink or it is mounted from another file system.

Solution: Reason 1: Move the /var/sadm directory into the root (/) or /var file system.

Cause: Reason 2: The /var/sadm/softinfo/INST_RELEASE file is missing.

Solution: Reason 2: Create a new INST_RELEASE file by using the following template:

```
OS=Solaris
VERSION=x
REV=0
```

x Is the version of Solaris software on the system

Cause: Reason 3: SUNWusr is missing from /var/sadm/softinfo.

Solution: Solution 3: You need to do an initial installation. The Solaris software is not upgradable.
Couldn’t shut down or initialize the md driver

Solution:
- If not a mirror, comment out in the vsftab file.
- If a mirror, break the mirror and reinstall.

The upgrade fails because the Solaris installation program cannot mount a file system.

Cause: During an upgrade, the script attempts to mount all the file systems that are listed in the system’s /etc/vfstab file on the root (/) file system that is being upgraded. If the installation script cannot mount a file system, it fails and exits.

Ensure that all file systems in the system’s /etc/vfstab file can be mounted. Comment out any file systems in the /etc/vfstab file that cannot be mounted or that might cause the problem so that the Solaris suninstall program does not try to mount them during the upgrade. Any system-based file systems that contain software to be upgraded (for example, /usr) cannot be commented out.

The upgrade fails

Description: The system does not have enough space for the upgrade.

Cause: Check Chapter 5 for the space problem and see if you can fix it without using auto-layout to reallocate space.

Problems upgrading mirrored roots

Solution: If you have problems upgrading when using Solaris Volume Manager mirrored roots, see “Troubleshooting Solaris Volume Manager” 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 341.

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.
      
      ```bash
      # lumount inactive_boot_environment_name mount_point
      ```
      
      For example:
      
      ```bash
      # lumount solaris8 /.alt.12345
      ```

   b. Change to the directory that contains the `vfstab` file, for example:
      
      ```bash
      # cd /.alt.12345/etc
      ```

   c. Make a copy of the inactive boot environment's `vfstab` file, for example:
      
      ```bash
      # cp vfstab vfstab.501
      ```

   d. In the copied `vfstab`, comment out all Veritas file system entries, for example:
      
      ```bash
      # 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 `vfstab` file comment lines.

   e. Copy the changed `vfstab` file, for example:
      
      ```bash
      # cp vfstab.novxfs vfstab
      ```

   f. Change directories to the inactive boot environment’s system file, for example:
      
      ```bash
      # cd /.alt.12345/etc
      ```

   g. Make a copy of the inactive boot environment’s system file, for example:
      
      ```bash
      # cp system system.501
      ```

   h. Comment out all "forceload:" entries that include `drv/vx`.
      
      ```bash
      # sed '/forceload:/ system.novxfs > system < drv/vx/^
      ```
      
      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:
   ```bash
   # cd /alt.12345/etc
   ```

j. Create the Veritas `install-db` file, for example:
   ```bash
   # touch vx/reconfig.d/state.d/install-db
   ```

k. Unmount the inactive boot environment.
   ```bash
   # lvmount 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 378.

5. Shut down the system.
   ```bash
   # init 0
   ```

6. Boot the inactive boot environment in single-user mode:
   ```bash
   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:
      ```bash
      # pkgrm VRTSvmsa
      ```
   
   b. Change directories to the Veritas packages.
      ```bash
      # cd /location_of_Veritas_software
      ```
   
   c. Add the latest Veritas packages to the system:
      ```bash
      # pkgadd -d 'pwd' VRTSvxvm VRTSvmsa VRTSvmdoc VRTSvmmvman VRTSvmdev
      ```

8. Restore the original `vfstab` and system files:
   ```bash
   # cp /etc/vfstab.original /etc/vfstab
   # cp /etc/system.original /etc/system
   ```

9. Reboot the system.
   ```bash
   # init 6
   ```
x86: Service Partition Not Created by Default on Systems With No Existing Service Partition

If you install the Solaris 9 operating environment on a system that does not currently include a Service partition, the installation program might not create a Service partition by default. If you want to include a Service partition on the same disk as the Solaris partition, you must recreate the Service partition before you install the Solaris 9 operating environment.

If you installed the Solaris 8 2/02 operating environment on a Sun LX50 system, the installation program might not have preserved the Service partition. If you did not manually edit the fdisk boot partition layout to preserve the Service partition, the installation program deleted the Service partition during the installation.

**Note** – If you did not specifically preserve the Service partition when you installed the Solaris 8 2/02 operating environment, you might not be able to recreate the Service partition and upgrade to the Solaris 9 operating environment.

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

- To use the Solaris Web Start installation program to install from the Solaris 9 Installation CD, follow these steps.
  1. Delete the contents of the disk.
  2. Before you install, create the Service partition by using the Sun LX50 Diagnostics CD.
     For information on how to create the Service partition, see the *Sun LX50 Server User’s Manual* and the Sun LX50 Knowledge Base at http://cobalt-knowledge.sun.com
  3. Insert the Solaris 9 Installation CD in the CD-ROM drive.
  4. Begin installing the Solaris 9 operating environment.

When the installation program detects the Service partition, the following message is displayed.

The default layout for the bootdisk is one x86 Boot partition and a Solaris partition on the remaining space. The Service fdisk partition, if one exists, is also preserved by default.

Select one of the following to continue:

1) Use the default layout
2) Run fdisk to manually edit the disk
3) Exit

Please make a selection: [?]
5. Type 1 to use the default layout.
The installation program preserves the Service partition and creates the x86 boot partition and the Solaris partition.

**Note** – The Solaris Web Start installation program creates the x86 boot partition by removing 10 Mbytes from the Solaris fdisk partition. This utility prevents any existing fdisk partitions from being altered. Do not create this partition manually.

6. Complete the installation.

- To install from a network installation image or from the Solaris 9 DVD over the network, follow these steps.
  1. Delete the contents of the disk.
  2. Before you install, create the Service partition by using the Sun LX50 Diagnostics CD.

  3. Boot the system from the network.

     The Customize fdisk Partitions screen is displayed.
  4. To load the default boot disk partition layout, click Default.

     The installation program preserves the Service partition and creates the x86 boot partition and the Solaris partition.

- To use the suninstall program to install from the Solaris 9 Software 1 of 2 CD or from a network installation image on a boot server, follow these steps.
  1. Delete the contents of the disk.
  2. Before you install, create the Service partition by using the Sun LX50 Diagnostics CD.

  3. Boot the system.

     The installation program prompts you to choose a method for creating the Solaris partition.
  4. Select the Use rest of disk for Solaris partition option.

     The installation program preserves the Service partition and creates the Solaris partition.
  5. Complete the installation.
Installing or Upgrading Remotely (Tasks)

This appendix describes how to use the Solaris Web Start program to install or upgrade to the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive.

**Note** – If you are installing or upgrading the Solaris operating environment on a multi-domain server, refer to the system controller or system service processor documentation before beginning the installation process.

**SPARC: Using the Solaris Web Start Program to Install or Upgrade From a Remote DVD-ROM or CD-ROM**

If you want to install the Solaris operating environment on a machine or domain that does not have a directly attached DVD-ROM or CD-ROM drive, you can use a drive that is attached to another machine. Both machines must be connected to the same subnet. Use the following instructions to complete the installation.
SPARC: To Install or Upgrade From a Remote DVD-ROM and CD-ROM

Note – This procedure assumes that the system is running the Volume Manager. If you are not using the Volume Manager to manage media, refer to System Administration Guide: 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 SPARC Platform Edition Installation CD in the drive. The Volume Manager mounts the disc.

3. On the *remote system*, change directories to the DVD or CD where the *add_install_client* command is located.
   - For DVD media, type:
     ```bash
     remote system# cd /cdrom/cdrom0/s0/Solaris_9/Tools
     ```
   - For CD media, type:
     ```bash
     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:
     ```bash
     remote system# ./add_install_client \\n     client_system_name arch
     ```
   - For CD media, type:
     ```bash
     remote system# ./add_install_client -s remote_system: \\
     /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
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.

Packaging Requirements Overview

The following documentation references provide background information on packaging requirements.

- For the custom JumpStart program and Solaris Live Upgrade to work properly, packages must comply with the SVR4 packaging requirements. The Application Packaging Developer’s Guide provides more specific information on packaging requirements and definitions of terminology. See especially the chapter: “Advanced Package Creation Techniques” in Application Packaging Developer’s Guide.

- For basic information on adding and removing packages and the installation administration file, see “Managing Software (Overview)” in System Administration Guide: Basic Administration. Also see the specific man pages.

- For detailed information on the commands referenced in this appendix, see the man pages, dircmp(1), fssnap(1M), ps(1), or truss(1).

Table C–1 lists information that applies to either Solaris Live Upgrade or the custom JumpStart program in this document.
Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements

An inactive boot environment is a copy of the operating environment, not the currently running system. A package to be used by Live Upgrade or the custom JumpStart program must follow these requirements:

- Enable a custom JumpStart installation or upgrade with no user interaction
- Make no modification of the current running system, which is necessary when using Solaris Live Upgrade

The following list explains the requirements for inactive boot environment compliance.

- For an installation of an operating system to be successful, packages must recognize and correctly respect inactive boot environment specifiers. Packages can include absolute paths in their pkgmap file (package map). If these files exist, they are written relative to the -R option of the pkgadd command. Packages that contain both absolute and relative (relocatable) paths can be installed to an alternative root (/) as well. $PKG_INSTALL_ROOT is prepended to both absolute and relocatable files so all paths are resolved properly when being installed by pkgadd.
- Packages being installed using the pkgadd -R option or being removed using the pkgrm -R option must not alter the currently running system.
Any procedure scripts that are included in the packages being installed with the pkgadd command -R option or being removed using the pkgrm command -R option must not alter the currently running system. Any installation scripts that you provide must reference any directory or file prefixed with the $PKG_INSTALL_ROOT variable. The package must write all directories and files with the $PKG_INSTALL_ROOT prefix. The package must not remove directories without a $PKG_INSTALL_ROOT prefix. Table C–2 provides examples of correct scripting syntax.

**TABLE C–2 Examples of Installation Script Syntax**

<table>
<thead>
<tr>
<th>Script Type</th>
<th>Correct Syntax</th>
<th>Incorrect Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bourne shell &quot;if&quot; statement fragments</td>
<td><code>if [ -f ${PKG_INSTALL_ROOT}/etc/myproduct.conf ] ; then</code></td>
<td><code>if [ -f /etc/myproduct.conf ] ; then</code></td>
</tr>
<tr>
<td>Removing a file</td>
<td><code>/bin/rm -f ${PKG_INSTALL_ROOT}/etc/myproduct.conf</code></td>
<td><code>/bin/rm -f /etc/myproduct.conf</code></td>
</tr>
<tr>
<td>Changing a file</td>
<td><code>echo &quot;test=no&quot; &gt; ${PKG_INSTALL_ROOT}/etc/myproduct.conf</code></td>
<td><code>echo &quot;test=no&quot; &gt; /etc/myproduct.conf</code></td>
</tr>
</tbody>
</table>

**Differences Between $PKG_INSTALL_ROOT and $BASEDIR Overview**

$PKG_INSTALL_ROOT is the location of the root (/) file system of the machine to which you are adding the package. The location is set to the -R argument of the pkgadd command. For example, if the following command is invoked, then $PKG_INSTALL_ROOT becomes /a during the installation of the package.

```
# pkgadd -R /a SUNWvxvm
```

$BASEDIR points to the relocatable base directory into which relocatable package objects are installed. Only relocatable objects are installed here. Non-relocatable objects (those that have absolute paths in the pkgmap file) are always installed relative to the inactive boot environment, but not relative to the $BASEDIR in effect. If a package has no relocatable objects, then the package is said to be an absolute package (or non-relocatable), and $BASEDIR is undefined and not available to package procedure scripts.

For example, suppose a package’s pkgmap file has two entries:

```
1 f none /sbin/ls 0555 root sys 3541 12322 1002918510
1 f none /sbin/ls2 0555 root sys 3541 12322 2342423332
```

The pkginfo file has a specification for $BASEDIR:
BASEDIR=/opt

If this package is installed with the following command, then ls is installed in /a/opt/sbin/ls, but ls2 is installed as /a/sbin/ls2.

# pkgadd -R /a SUNWtest

---

**Solaris Live Upgrade Inactive Boot Environment Compliance**

When using Solaris Live Upgrade and creating a new boot environment, avoid problems by following these guidelines.

- Your package procedure scripts must be independent of the current active operating environment. Procedure scripts define actions that occur at particular points during package installation and removal. Four procedure scripts can be created with these predefined names: preinstall, postinstall, preremove, and postremove. Package procedure scripts must be independent of the currently active operating environment because an inactive boot environment could be switched by using Solaris Live Upgrade.

- These scripts must not start or stop any processes or depend on the output of commands such as ps or truss, which are operating-system dependent and report information about the currently running system.

- Procedure scripts are free to use other standard UNIX commands such as expr, cp, and ls and other commands that facilitate shell scripting. But, the inactive boot environment must not be modified except within the rules outlined in the section, “Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements” on page 464.

- All scripts must be written in Bourne shell (/bin/sh). Bourne shell is the interpreter used by the pkgadd command to execute the procedure scripts.

- Package procedure scripts must not invoke commands that were added in the 2.6 and later releases. For example, package procedure scripts cannot invoke the pgrep command. Since the 2.6 release, many commands have had additional features added. Package procedure scripts must not use any command options that did not exist in the 2.6 release. For example, the -f option of the umount command was added in the Solaris 7 release. To verify that a specific command or option is supported in the Solaris 2.6 release, see the Solaris 2.6 Reference Manual AnswerBook on http://docs.sun.com.

- All packages must pass pkgchk validation. After a package is created and before it is installed, it must be checked with the following command.

  # pkgchk -d dir_name pkg_name
**dir_name**

Specifies the name of the directory where the package resides

**pkg_name**

Specifies the name of the package

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

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

No errors should be displayed.

After a package is created, it must be tested by installing it in an inactive boot environment location using the `-R dir_name` option to `pkgadd`. After the package is installed, it must be checked for correctness using `pkgchk`, as in this example.

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

No errors should be displayed.

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

These requirements for creating, modifying, and deleting files can be verified using a variety of commands. For example, the `dircmp` or `fssnap` commands can be used to verify that packages behave properly. Also, the `ps` command can be used for testing daemon compliance by making sure daemons are not stopped or started by the package. The `truss`, `pkgadd -v`, and `pkgrm` commands can test runtime package installation compliance, but might not work in all situations. In the following example, the `truss` command strips out all read-only, non-$TEMPDIR access and shows only non-read-only access to paths that do not lie within the specified inactive boot environment.

```bash
# TEMPDIR=/a; export TEMPDIR
# truss -t open /usr/sbin/pkgadd -R ${TEMPDIR} SUNWvxvm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v "open(${TEMPDIR})"
```

For detailed information on the commands referenced in this section, see the man pages `dircmp(1)`, `fssnap(1M)`, `ps(1)`, `truss(1)`, `pkgadd(1M)`, `pkgchk(1M)`, or `pkgrm(1M)`. 
Upgrade Compliance With the Custom JumpStart Program

Custom JumpStart program compliance ensures that packages can be added and removed while being part of the traditional Solaris installation utilities, which are the following:

- The custom JumpStart program
- Solaris suninstall program
- Solaris Web Start installation method

The custom JumpStart program compliance also ensures that the package can participate in Solaris upgrades. To be custom JumpStart program-compliant, a package also must follow inactive boot environment requirements that are outlined in “Custom JumpStart Program and Solaris Live Upgrade Inactive Boot Environment Requirements” on page 464.

To be able to use the custom JumpStart program effectively, packages must be added or removed without the user being prompted for information. To prevent user interaction, set up a new administration file with the pkgadd command -a option. The -a option defines an installation administration file to be used in place of the default administration file. Using the default file might result in the user being prompted for more information. You can create an administration file that indicates to pkgadd that it should bypass these checks and install the package without user confirmation. For details, see the man pages, admin(4) or pkgadd(1M).

The following examples show how to use the pkgadd administration file.

- If no administration file is provided, pkgadd uses /var/sadm/install/admin/default. Using this file might result in user interaction.
  
  # pkgadd

- If a relative administration file is provided on the command line, pkgadd looks in /var/sadm/install/admin for the file name and uses it. In this example, the relative administration file is named nocheck and pkgadd looks for /var/sadm/install/admin/nocheck.

  # pkgadd -a nocheck

- If an absolute file is provided pkgadd uses it. In this example, pkgadd looks in /tmp for the nocheck administration file.

  # pkgadd -a /tmp/nocheck
The following is an example of an installation administration file that requires very little user interaction with the `pkgadd` utility. Unless the package requires more space than is available on the system, the `pkgadd` utility uses this file and installs the package without prompting the user for more information.

```bash
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
space=ask
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```
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 suuninstall 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 472.

After you perform the analysis, refer to “To Review the Patch Analyzer Output” on page 472 for detailed information about the patch analysis results.
To Run the `analyze_patches` Script

1. Change to the **Misc** directory.
   - **SPARC:** If the image is located on locally mounted media, type:
     ```bash
     # cd /cdrom/sol_9_Update_sparc/s0/Solaris_9/Misc
     ```
     In this command, *Update* is the actual Update identifier.
   - **x86:** If the image is located on locally mounted media, type:
     ```bash
     # cd /cdrom/sol_9_Update_x86/s2/Solaris_9/Misc
     ```
     In this command, *Update* is the actual Update identifier.
   - If the image is available on an NFS file system, type:
     ```bash
     # cd /NFS_mount_directory/Solaris_9/Misc
     ```

2. Run the `analyze_patches` script:
   ```bash
   # ./analyze_patches [-R rootdir] [-N netdir] [-D databasedir]
   ```
   - **rootdir** is the root of the installed system. The default is `/.
   - **netdir** is the path to the root of the OS image to be installed. The default is `/cdrom/cdrom0`. `netdir` is the path to the directory that contains the `solaris_9` directory. You must use this option if you are running the `patch_analyzer` from an NFS mount point.
   - **databasedir** If the script is invoked from a directory other than the **Misc/ directory in the OS image, the program cannot find the database it uses for patch analysis. Use the `-D` option to supply the path to the database. Without this database, which is located in `Solaris_9/Misc/database` on the OS image, the script does not work properly.

To Review the Patch Analyzer Output

After you perform the analysis, use these steps to review the output.

1. **Review the output of the Patch Analyzer.**
   The Patch Analyzer provides a list of patches that will be removed, downgraded, accumulated, or obsoleted by other patches. Patch accumulations are similar to patch upgrades. The accumulated patch is removed and its fixes are delivered by a new
patch. Messages such as the following are shown:

Patch 105644-03 will be removed.
Patch 105925 will be downgraded from -02 to -01.
Patch 105776-01 will be accumulated/obsoleted by patch 105181-05.

If the Patch Analyzer program does not provide a list, no action is taken against any patches that were previously installed on your system.

2. Decide if the patch replacements and deletions are acceptable.
   - If yes, upgrade the system.
   - If no, do not upgrade the system.

At an update release, instead of upgrading, you can use the Solaris 9 Maintenance Update release to apply only patches to your system.

---

**Note** – The Solaris 9 Maintenance Update is available on CD and by download. Instructions for applying patches are provided in the *Solaris 9 Maintenance Update Installation Guide*.  

---
x86: Preparing to Boot From the Solaris 9 Device Configuration Assistant or the Network (Tasks)

This appendix describes the following topics.

- “x86: Copying the Boot Software to a Diskette” on page 475
- “x86: PXE Network Boot” on page 477

x86: Copying the Boot Software to a Diskette

The Solaris Device Configuration Assistant is a program that enables you to perform various hardware configuration and booting tasks. The Solaris 9 Device Configuration Assistant image is found in the Tools directory of either the Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 2 of 2 x86 Platform Edition CD. Use the following procedure to copy the boot image to a 3.5 diskette.

Note – You can boot directly from DVD or CD media or by using a net image with PXE. For information on these methods of booting, see “x86: Accessing the Solaris 9 Device Configuration Assistant and PXE” on page 31.
x86: To Copy the Boot Software to a Diskette

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

1. Log in as superuser on an x86 system to which a diskette drive is attached.

2. On the system with the DVD-ROM or CD-ROM drive, insert the Solaris 9 x86 Platform Edition DVD or the Solaris 9 Software 2 of 2 x86 Platform Edition CD in the drive.
   The Volume Manager mounts the disc.

3. Change to the directory that contains the boot image.
   
   - For DVD media, type:
     
     # cd /cdrom/sol_9_x86/s2/Solaris_9/Tools
   
   - For CD media, type:
     
     # cd /cdrom/sol_9_x86/Solaris_9/Tools

4. Insert a blank diskette or a diskette that can be overwritten in the diskette drive.

5. Notify Volume Manager to check for new media.
   
   # volcheck

6. Format the diskette:
   
   Caution – Formatting erases all data on the diskette.

   # fdformat -d -U

7. Copy the file to the diskette.
   
   # dd if=d1_image of=/vol/dev/aliases/floppy0 bs=36k

8. Eject the diskette by typing eject floppy at the command line, and then manually ejecting the diskette from the drive.
x86: PXE Network Boot

PXE network boot is a "direct" network boot. No boot media is required on the client system. PXE network boot is available only for devices that implement the Intel Preboot Execution Environment specification. To determine if your system supports PXE network boot, see your hardware manufacturer’s documentation.

The Solaris boot diskette is still available for systems that do not support this feature. You can get the boot diskette image from http://soldc.sun.com/support/drivers/dca_diskettes.

Enable PXE network boot on the client system by using the BIOS setup program in the system BIOS, the network adapter BIOS, or both. On some systems you must also adjust the boot device priority list so that network boot is attempted before booting from other devices. See the manufacturer’s documentation for each setup program, or watch for setup program entry instructions during boot.

Some PXE-capable network adapters have a feature that enables PXE boot if you type a particular keystroke in response to a brief boot-time prompt. This is ideal when using PXE for an install boot on a system that normally boots from the disk drive because you do not have to modify the PXE settings. If your adapter does not have this feature, disable PXE in the BIOS setup when the system reboots after installation, and the system will boot from the disk drive.

Some early versions of PXE firmware cannot boot the Solaris system. If you have one of these older versions, your system will be able to read the PXE network bootstrap program from a boot server, but the bootstrap will not transmit packets. If this happens, upgrade the PXE firmware on the adapter. Obtain firmware upgrade information from the adapter manufacturer’s web site. Refer to the elx1(7D) and iprb(7D) man pages for more information.
Solaris 9 Updates

This appendix describes changes to this book for Solaris 9 update releases.

Solaris 9 9/02 Release.

Minor changes were made to fix a bug.

Solaris 9 12/02 Release

The following list describes new features and changes to this book for the Solaris 9 12/02 release.

- When using the Solaris Flash installation feature, several new procedures and examples have been added.
  - The `flar create` command is used to create a Solaris Flash archive. The command has been updated with new options that increase your flexibility to define archive contents when creating an archive. You now can exclude more than one file or directory. From an excluded directory, you can restore a subdirectory or file. This feature is useful when you want to exclude large data files that you do not want cloned. For information on how to use these options, see “Customizing an Archive’s Files and Directories” on page 164.
  - New procedures for splitting, merging, or extracting information about the archive have been added. For information on these procedures, see “Administering Solaris Flash Archives” on page 186.
The Solaris installation programs now support LDAP Version 2 profiles, which enable you to configure your system to use a proxy credential level. During the Solaris Web Start or Solaris suninstall program installation programs, you can specify the LDAP proxy bind distinguished name and proxy bind password. With any installation method, you can preconfigure LDAP before installation by using the `proxy_dn` and `proxy_password` keywords in the `sysidcfg` file. For information on preconfiguring these profiles, see “Preconfiguring With the `sysidcfg` File” on page 57.

When using Solaris Live Upgrade, new procedures and examples have been added, which are the following:

- Creating a profile
- Testing the profile
- Using the profile to upgrade or install a Solaris Flash archive

For more information, see Chapter 33

This book now contains procedures and descriptions that relate to x86 based systems.

Minor changes were made to fix bugs.

### Solaris 9 4/03 Release

The following list describes new features and changes to this book for the Solaris 9 4/03 release.

- The Solaris Flash installation feature provides new enhancements for this Solaris release.

- A Solaris Flash installation can now update a clone system with minor changes. If you have a clone system and want to update it with minor changes, you can create a differential archive that contains only the differences between two images, the original master image and an updated master image. When you update a clone system with a differential archive, only the files specified in the differential archive are changed. The installation is restricted to clone systems that contain software consistent with the original master image. You use the custom JumpStart installation method to install an differential archive on a clone system.

- Special scripts can now be run for configuration of the master or clone or run to validate the archive. These scripts enable you to do the following tasks.

  - Configure applications on clone systems. You can use a custom JumpStart script for some uncomplicated configurations. For more complicated configurations, special configuration file processing might be necessary on the master system or before or after installation on the clone system. Also,
local preinstallation and postinstallation scripts can reside on the clone and protect local customizations from being overwritten by the Solaris Flash software.

- Identify non-clonable, host-dependent data that enables you to make the flash archive host independent. Host independence is enabled by modifying such data or excluding it from the archive. An example of host-dependent data is a log file.
- Validate software integrity in the archive during creation
- Validate the installation on the clone system
- In the Solaris 9 4/03 operating environment, the Solaris Web Start and suninstall installation programs use a new default boot-disk partition layout to accommodate the Service partition on x86–based systems. If your system currently includes a Service partition, the new default boot-disk partition layout enables you to preserve this partition.
  
  For more information, see “x86: Change in Default Boot-Disk Partition Layout” on page 30
- Minor changes were made to fix bugs.
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>archive</td>
<td>A file that contains a collection of files that were copied from a master system. The file also contains identification information about the archive, such as a name and the date that you created the archive. After you install an archive on a system, the system contains the exact configuration of the master system. An archive could be a differential archive which is Solaris Flash archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files specified and is restricted to systems that contain software consistent with the unchanged master image.</td>
</tr>
<tr>
<td>arrow keys</td>
<td>One of the four directional keys on the numeric keypad.</td>
</tr>
<tr>
<td>begin script</td>
<td>A user-defined Bourne shell script, specified within the rules file, that performs tasks before the Solaris software is installed on the system. You can use begin scripts only with custom JumpStart installations.</td>
</tr>
<tr>
<td>boot</td>
<td>To load the system software into memory and start it.</td>
</tr>
<tr>
<td>boot environment</td>
<td>A collection of mandatory file systems (disk slices and mount points) that are critical to the operation of the Solaris operating environment. These disk slices might be on the same disk or distributed across multiple disks. The active boot environment is the one that is currently booted. Exactly one active boot environment can be booted. An inactive boot environment is not currently booted, but can be in a state of waiting for activation on the next reboot.</td>
</tr>
<tr>
<td>boot server</td>
<td>A server system that provides client systems on the same network subnet with the programs and information that they need to start. A</td>
</tr>
</tbody>
</table>
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 Solaris Flash archive. The clone system has the same installation configuration as the master system.

**cluster**

A logical collection of packages (software modules). The Solaris 9 software is divided into *software groups*, which are each composed of clusters and *packages*.

**command line**

A string of characters that begins with a command, often followed by arguments, including options, file names, and other expressions, and terminated by the end-of-line character.

**Core software group**

A software group that contains the minimum software that is required to boot and run the Solaris operating environment on a system. Core includes some networking software and the drivers that are required to run the Common Desktop Environment (CDE) desktop. Core does not include the CDE software.

**critical file systems**

File systems that are required by the Solaris operating environment. When you use Solaris Live Upgrade, these file systems are separate mount points in the *vfstab* of the active and inactive boot environments. Examples are *root* (/), *usr*, *var*, or *opt*. These file systems are always copied from the source to the inactive boot environment.

**custom JumpStart**

A type of installation in which the Solaris 9 software is automatically installed on a system based on a user-defined profile. You can create customized profiles for different types of users and systems. A custom JumpStart installation is a JumpStart installation you create.

**custom probes file**

A file, which must be located in the same JumpStart directory as the *rules* file, is a Bourne shell script that contains two types of functions: probe and comparison. Probe functions gather the information you want or do the actual work and set a corresponding *SI_environment* variable you define. Probe functions become probe keywords. Comparison functions call a corresponding probe function, compare
the output of the probe function, and return 0 if the keyword matches or 1 if the keyword doesn’t match. Comparison functions become rule keywords. See also rules file.

**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.

**differential archive**

A Solaris Flash archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files specified and is restricted to systems that contain software consistent with the unchanged master image.

**disc**

An optical disc, as opposed to a magnetic disk, which recognizes the common spelling that is used in the compact disc (CD) market. For example, a CD-ROM or DVD-ROM is an optical disc.

**disk**

A round platter, or set of platters, of a magnetized medium that is organized into concentric tracks and sectors for storing data such as files. See also disc.

**disk configuration file**

A file that represents a structure of a disk (for example, bytes/sector, flags, slices). Disk configuration files enable you to use pfinstall from a single system to test profiles on different–size disks.

**diskless client**

A client on a network that relies on a server for all of its disk storage.

**domain**

A part of the Internet naming hierarchy. A domain represents a group of systems on a local network that share administrative files.

**domain name**

The name that is assigned to a group of systems on a local network that share administrative files. The domain name is required for the Network Information Service (NIS) database to work properly. A domain name consists of a sequence of component names that are separated by periods (for example: tundra.mpk.ca.us). As you read a domain name from left to right, the component names identify more general (and usually remote) areas of administrative authority.
<table>
<thead>
<tr>
<th><strong>End User Solaris Software Group</strong></th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entire Solaris Software Group</strong></td>
<td>A software group that contains the entire Solaris 9 release.</td>
</tr>
<tr>
<td><strong>Entire Solaris Software Group Plus OEM Support</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>EISA</strong></td>
<td>Extended Industry Standard Architecture. A type of bus on x86 based systems. EISA bus standards are “smarter” than ISA bus systems. Attached devices can be automatically detected when they have been configured with the “EISA configurator” program that was supplied with the system. See also ISA.</td>
</tr>
<tr>
<td><strong>/etc</strong></td>
<td>A directory that contains critical system configuration files and maintenance commands.</td>
</tr>
<tr>
<td><strong>/export</strong></td>
<td>A file system on an OS server that is shared with other systems on a network. For example, the /export file system can contain the root file system and swap space for diskless clients and the home directories for users on the network. Diskless clients rely on the /export file system on an OS server to boot and run.</td>
</tr>
<tr>
<td><strong>fallback</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>fdisk partition</strong></td>
<td>A logical partition of a disk drive that is dedicated to a particular operating system on x86 based systems. To install the Solaris software, you must set up at least one Solaris 9 fdisk partition on an x86 based system. x86 based systems allow up to four different fdisk partitions on a disk. These partitions can be used to hold individual operating systems. Each operating system must be located on a unique fdisk partition. A system can only have one Solaris fdisk partition per disk.</td>
</tr>
<tr>
<td><strong>file server</strong></td>
<td>A server that provides the software and file storage for systems on a network.</td>
</tr>
<tr>
<td><strong>file system</strong></td>
<td>In the SunOS™ operating system, a tree-structured network of files and directories that you can access.</td>
</tr>
<tr>
<td><strong>finish script</strong></td>
<td>A user-defined Bourne shell script, specified within the rules file, that performs tasks after the Solaris software is installed on the system, but before the system reboots. You use finish scripts with custom JumpStart installations.</td>
</tr>
<tr>
<td><strong>format</strong></td>
<td>To put data into a structure or divide a disk into sectors for receiving data.</td>
</tr>
</tbody>
</table>
**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.

**initial installation**
An installation that overwrites the currently running software or initializes a blank disk.

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

**initial installation option**
An option that is presented by the Solaris Web Start program and the Solaris suninstall program that overwrites the disk(s) with a new version of Solaris. The initial installation option is presented for systems that can be upgraded. However, the disk(s) that contain the old version of Solaris software (including the local modifications) are overwritten if you choose the initial installation option.

**install server**
A server that provides the Solaris 9 DVD or CD images from which other systems on a network can install Solaris (also known as a media server). You can create an install server by copying the Solaris 9 DVD or CD images to the server's hard disk.

**IP address**
Internet protocol address. In TCP/IP, a unique 32-bit number that identifies each host in a network. An IP address consists of four numbers that are separated by periods (192.9.9.1, for example). Most often, each part of the IP address is a number between 0 and 255. 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."
### Table: Class and IP Address Ranges

<table>
<thead>
<tr>
<th>Class</th>
<th>Range (xxx is a number 0 to 255)</th>
<th>Number of Available IP Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>1.xxx.xxx.xxx - 126.xxx.xxx.xxx</td>
<td>Over 16 million</td>
</tr>
<tr>
<td>Class B</td>
<td>128.0.xxx.xxx - 191.255.xxx.xxx</td>
<td>Over 65,000</td>
</tr>
<tr>
<td>Class C</td>
<td>192.0.0.xxx - 223.255.255.xxx</td>
<td>256</td>
</tr>
</tbody>
</table>

**IPv6**

IPv6 is a version (version 6) of Internet Protocol (IP) that is designed to be an evolutionary step from the current version, IPv4 (version 4). Deploying IPv6, by using defined transition mechanisms, does not disrupt current operations. In addition, IPv6 provides a platform for new Internet functionality.

IPv6 is described in more detail in “IPv6 (Overview)” in *System Administration Guide: IP Services*.

**ISA**

Industry Standard Architecture. A type of bus found in x86 based systems. ISA bus systems are “dumb” and provide no mechanism the system can use to detect and configure devices automatically. See also EISA.

**job**

A user-defined task to be completed by a computer system.

**JumpStart directory**

When you use a profile diskette for custom JumpStart installations, the JumpStart directory is the root directory on the diskette that contains all the essential custom JumpStart files. When you use a profile server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files.

**JumpStart installation**

A type of installation in which the Solaris 9 software is automatically installed on a system by using the factory-installed JumpStart software.

**Kerberos**

A network authentication protocol that uses strong, secret-key cryptography to enable a client and server to identify themselves to each other over an insecure network connection.

**LDAP**

Lightweight Directory Access Protocol is a standard, extensible directory access protocol that is used by LDAP naming service clients and servers to communicate with each other.

**locale**

A geographic or political region or community that shares the same language, customs, or cultural conventions (English for the U.S. is en_US, and English for the U.K. is en_UK).

**manifest section**

A section of a Solaris Flash archive that is used to validate a clone system. The manifest section lists the files on a system to be retained,
added to, or deleted from the clone system. This section is informational only. The sections lists the files in an internal format and cannot be used for scripting.

**master system**
A system that you use to create a Solaris Flash archive. The system configuration is saved in the archive.

**media server**
See install server.

**metadevice**
See volume.

**miniroot**
The smallest possible bootable Solaris root file system. A miniroot contains a kernel and just enough software to install the Solaris environment on a hard disk. The miniroot is the file system that is copied to a machine in the initial installation.

**mount**
The process of accessing a directory from a disk that is attached to a machine that is making the mount request or a remote disk on a network. To mount a file system, you need a mount point on the local system and the name of the file system to be mounted (for example, /usr).

**mount point**
A workstation directory to which you mount a file system that exists on a remote machine.

**name server**
A server that provides a name service to systems on a network.

**name service**
A distributed network database that contains key system information about all the systems on a network so that the systems can communicate with each other. With a name service, the system information can be maintained, managed, and accessed on a network-wide basis. Without a name service, each system has to maintain its own copy of the system information in the local /etc files. Sun supports the following name services: LDAP, NIS, and NIS+.

**networked systems**
A group of systems (called hosts) that are connected through hardware and software, so that they can communicate and share information. Referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.

**network installation**
A way to install software over the network—from a system with a CD-ROM or DVD-ROM drive to a system without a CD-ROM or DVD-ROM drive. Network installations require a name server and an install server.

**NIS**
The SunOS™ 4.0 (minimum) Network Information Service. A distributed network database that contains key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.

**NIS+**
The SunOS 5.0 (minimum) Network Information Service. NIS+ replaces NIS, the SunOS 4.0 (minimum) Network Information Service.
<table>
<thead>
<tr>
<th><strong>non-networked systems</strong></th>
<th>Systems that are not connected to a network or do not rely on other systems.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>/opt</strong></td>
<td>A file system that contains the mount points for third-party and unbundled software.</td>
</tr>
<tr>
<td><strong>OS server</strong></td>
<td>A system that provides services to systems on a network. To serve diskless clients, an OS server must have disk space set aside for each diskless client’s root file system and swap space (/export/root, /export/swap).</td>
</tr>
<tr>
<td><strong>package</strong></td>
<td>A collection of software that is grouped into a single entity for modular installation. The Solaris 9 software is divided into software groups, which are each composed of clusters and packages.</td>
</tr>
<tr>
<td><strong>panel</strong></td>
<td>A container for organizing the contents of a window, dialog box, or applet. The panel might collect and confirm user input. Panels might be used by wizards and follow an ordered sequence to fulfill a designated task.</td>
</tr>
<tr>
<td><strong>patch analyzer</strong></td>
<td>A script that you can run manually or as part of the Solaris suninstall program or the Solaris Web Start program. The patch analyzer performs an analysis on your system to determine which (if any) patches will be removed by upgrading to a Solaris 9 Update.</td>
</tr>
<tr>
<td><strong>platform group</strong></td>
<td>A vendor-defined grouping of hardware platforms for the purpose of distributing specific software. Examples of valid platform groups are i86pc and sun4u.</td>
</tr>
<tr>
<td><strong>platform name</strong></td>
<td>The output of the <code>uname -i</code> command. For example, the platform name for the Ultra 60 is SUNW,Ultra-60.</td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>Software that automatically saves the state of a system and turns it off after it is idle for 30 minutes. When you install the Solaris software on a system that complies with Version 2 of the U.S. Environmental Protection Agency’s Energy Star guidelines—a sun4u SPARC system, for example—the Power Management software is installed by default. After a subsequent reboot, you are prompted to enable or disable the Power Management software. Energy Star guidelines require that systems or monitors automatically enter a “sleep state” (consume 30 watts or less) after the system or monitor becomes inactive.</td>
</tr>
<tr>
<td><strong>probe keyword</strong></td>
<td>A syntactical element that extracts attribute information about a system when using the custom JumpStart method to install. A probe keyword does not require you to set up a matching condition and run a profile as required for a rule. See also rule.</td>
</tr>
<tr>
<td><strong>profile</strong></td>
<td>A text file that defines how to install the Solaris software when using the custom JumpStart method. For example, a profile defines which software group to install. Every rule specifies a profile that defines...</td>
</tr>
</tbody>
</table>
how a system is to be installed when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule. See also *rules file*.

**profile diskette**  
A diskette that contains all the essential custom JumpStart files in its root directory (JumpStart directory).

**profile server**  
A server that contains all the essential custom JumpStart files in a JumpStart directory.

**/ (root)**  
In a hierarchy of items, the one item from which all other items are descended. The root item has nothing above it in the hierarchy. / is the base directory from which all other directories stem, directly or indirectly. The root directory contains the directories and files critical for system operation, such as the kernel, device drivers, and the programs that are used to start (boot) a system.

**rule**  
A series of values that assigns one or more system attributes to a profile. A rule is used in a custom JumpStart installation.

**rules file**  
A text file that contains a rule for each group of systems or single systems that you want to install automatically. Each rule distinguishes a group of systems, based on one or more system attributes. The rules file links each group to a profile, which is a text file that defines how the Solaris 9 software is to be installed on each system in the group. A rules file is used in a custom JumpStart installation. See also *profile*.

**rules.ok file**  
A generated version of the rules file. The rules.ok file is required by the custom JumpStart installation software to match a system to a profile. You must use the check script to create the rules.ok file.

**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 /export/home and /swap. These file systems are shared between the active and inactive boot environment when you use Solaris Live Upgrade. Shareable file systems contain the same mount point in the vfstab in both the active and inactive boot environments. Updating shared files in the active boot environment also updates data in the inactive boot environment. Shareable file systems are shared by default, but you can specify a destination slice, and then the file systems are copied.

**slice**  
The unit into which the disk space is divided by the software.

**software group**  
A logical grouping of the Solaris software (clusters and packages). During a Solaris installation, you can install one of the following software groups: Core, End User Solaris Software, Developer Solaris Software, or Entire Solaris Software, and for SPARC systems only, Entire Solaris Software Group Plus OEM Support.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 9 DVD or CD images</td>
<td>The Solaris 9 software that is installed on a system, which you can access on the Solaris 9 DVDs or CDs or an install server’s hard disk to which you have copied the Solaris 9 DVD or CD images.</td>
</tr>
<tr>
<td>Solaris Flash</td>
<td>A Solaris installation feature that enables you to create an archive of the files on a system, known as the master system. You can then use the archive to install other systems, making the other systems identical in their configuration to the master system. See also archive.</td>
</tr>
<tr>
<td>Solaris suninstall program</td>
<td>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.</td>
</tr>
<tr>
<td>Solaris Live Upgrade</td>
<td>An upgrade method that enables a duplicate boot environment to be upgraded while the active boot environment is still running, thus eliminating downtime of the production environment.</td>
</tr>
<tr>
<td>Solaris Web Start program</td>
<td>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.</td>
</tr>
<tr>
<td>standalone</td>
<td>A computer that does not require support from any other machine.</td>
</tr>
<tr>
<td>subnet</td>
<td>A working scheme that divides a single logical network into smaller physical networks to simplify routing.</td>
</tr>
<tr>
<td>subnet mask</td>
<td>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.</td>
</tr>
<tr>
<td>swap space</td>
<td>A slice or file that temporarily holds the contents of a memory area till it can be loaded back into memory. Also known as the /swap or swap file system.</td>
</tr>
<tr>
<td>sysidcfg file</td>
<td>A file in which you specify a set of special system configuration keywords that preconfigure a system.</td>
</tr>
<tr>
<td>time zone</td>
<td>Any of the 24 longitudinal divisions of the earth’s surface for which a standard time is kept.</td>
</tr>
<tr>
<td>unmount</td>
<td>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.</td>
</tr>
<tr>
<td>update</td>
<td>An installation, or to perform an installation, on a system that changes software that is of the same type. Unlike an upgrade, an update might downgrade the system. Unlike an initial installation, software of the same type that is being installed must be present before an update can occur.</td>
</tr>
<tr>
<td>upgrade</td>
<td>An installation that merges files with existing files and saves modifications where possible.</td>
</tr>
</tbody>
</table>
An upgrade of the Solaris operating environment merges the new version of the Solaris operating environment with the existing files on the system’s disk or disks. An upgrade saves as many modifications as possible that you have made to the previous version of the Solaris operating environment.

**upgrade option**
An option that is presented by the Solaris Web Start program and the Solaris suninstall program. The upgrade procedure merges the new version of Solaris with existing files on your disk(s). An upgrade also saves as many local modifications as possible since the last time Solaris was installed.

**/usr**
A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large /usr file system with a server rather than maintaining a local copy minimizes the overall disk space that is required to install and run the Solaris 9 software on a system.

**utility**
A standard program, usually furnished at no charge with the purchase of a computer, that does the computer’s housekeeping.

**/var**
A file system or directory (on standalone systems) that contains system files that are likely to change or grow over the life of the system. These files include system logs, vi files, mail files, and uucp files.

**volume**
A group of physical slices 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 DVD-ROMs, CD-ROMs, and diskettes.
Index

Numbers and Symbols
#
  in profiles, 225
  in rules files, 222
= (equal sign) in profile field, 236
! (exclamation mark) rule field, 223
&& (ampersands) rule field, 223
(/) file systems
  value set by JumpStart, 307

A
activating a boot environment for Live Upgrade, overview, 323
activation failure of boot environment for Live Upgrade, overview, 325
add_install_client, description, 123
add_install_client command
example
  boot server for DVD media, 93
  for boot server for CD media, 122
  same subnet for CD media, 122
  with DHCP for CD media, 121 122
  with DHCP for DVD media, 93
example for cross-platform CD media, 122
install server setup
  with CD media, 120
  with DVD media, 91
JumpStart directory access, 215
add_to_install_server, description, 123
adding
  clusters when upgrading, 293
  (Continued)
  dataless clients
    with CD media, 119
    with DVD media, 91
  files with a finish script, 238
  install server configuration
    with CD media, 120
    with DVD media, 91
locale.org_dir table entries, 65
packages and patches with a finish script, 238
packages from software groups, 305, 369
rules to rules file, 222
systems from network, 82, 96
alternative installation programs, 248
ampersands (&&) rule field, 223
AND rule field, 223
any
  probe keyword
    description and values, 312
  rule keyword
    description and values, 277, 311
arch probe keyword, 311
arch rule keyword, 277, 311
archive
  See also clone systems
  See also differential archive
  See also scripts
  See also Solaris Flash
administering, 186
compared to other installation methods, 35
compressing, 167
archive (Continued)
creating an archive
description, 161
differential, 180
examples, 177, 182
initial install, 176
customizing
description, 164
with scripts, 164
flar create command, 198
getting information, 188
installing
custom JumpStart, 257
description, 155
how to install, 185
installation methods, 167
JumpStart profile example, 228, 229
keywords
description, 193
identification section, 194
section_begin and section_end, 193
user-defined, 197
planning installation, 167
sections
archive cookie, description, 191
archive identification, description, 192
archives files, description, 193
creating, 173
description, 165 191
manifest, description, 192
summary, description, 192
user-defined, description, 192
updating a clone
description, 157
archive_location keyword, 283
auto_install_sample directory
check script, 234, 253
copying files to JumpStart directory, 214, 218, 221
set_root_pw finish script, 240 241

banner command, 124
begin.log file, 236
begin rule field, description, 223
begin scripts
creating derived profiles with, 236 237
overview, 235
permissions, 236
rule field, 223
site-specific installation programs, 248
boot: cannot open /kernel/unix message, 443
boot_device keyword, 290
boot disk partition layout, new default (x86 based systems), 30
boot environment, view status, 403
boot failure of boot environment for Live Upgrade, overview, 325
boot server
creating on a subnet with DVD, 88
creating on subnet with DVD media, 89
creating on subnet with CD media, 117
creating with DVD, example, 90
description, 78
requirement for network installation, 78
booting the system, resetting terminals and display first, 124
bootparams file
enabling JumpStart directory access, 216
updating, 448
Bourne shell scripts in rule fields, 223

C
-c option
pfinstall command, 232
add_install_client command, 121, 274
Can’t boot from file/device message, 443
cancel a Solaris Live Upgrade job, 394
cannot open /kernel/unix message, 443
CHANGE DEFAULT BOOT DEVICE message, 450
change name of Solaris Live Upgrade boot environment, 399
changing directories
image of Solaris 9 SPARC Platform Edition software on local disk, 218

B
-b option of setup_install_server command, 270
backslash in rules files, 222
backup_media keyword, 72, 288
bad blocks, 450
changing directories (Continued)

to image of Solaris 9 software on local disk, 214
to image of Solaris 9 x86 Platform Edition software on local disk, 220
to JumpStart directory, 234, 253

check script

custom_probes file validation, 253, 254
custom_probes.ok file creation, 253
derived profiles and, 237
rules file validation, 254
rules file validation, 233, 234
rules.ok file creation, 233
testing rules, 234, 254

client_arch keyword, 291
client_name, description, 121
client_root profile keyword, 291
clock gained xxx days message, 443

closest system

See also archive
description, 155

closest profile keyword
description and values, 292, 293, 369
elements, 226

color depth, preconfiguring, 57

commands for Solaris Live Upgrade, 413
comments
in profiles, 225
in rules files, 222

compare Live Upgrade boot environments, 395
configuring
creating disk configuration files, 243
Solaris Live Upgrade file systems, 343

copying
JumpStart directory files, 238
JumpStart installation files, 214, 218, 221
Solaris 9 Device Configuration Assistant x86
Platform Edition diskette, 219
Solaris Live Upgrade file systems, 392

Core Solaris Software Group, 292
Core System Support
description, 40
size, 41
CPUs (processors)
probe keywords, 311
rule keywords, 277, 311

creating
a boot environment for Live Upgrade, 316, 341
boot server on a subnet
tasks, CD media, 96
tasks, DVD media, 82
boot server on subnet
with CD media, 117
boot server on subnet with DVD media, 89
/etc/locale file, 64
custom_probes.ok file, 253
disk configuration files, 243
install server, 86
install server with CD media, 96, 97, 102
install server with cross-platform CD media, 110, 116
install server with DVD media, 81, 83
JumpStart directory
server, 213
local file systems, 297
profiles
derived, 236
description, 225
rules file, 222
rules.ok file, 253
rules.ok file, 233
Solaris Flash archives
customizing, 164
description, 161
planning, 163
tasks, 176
sysidcfg file, 61
UFS, 218
critical file systems, definition, 316
.cshrc file, 239
custom JumpStart installation, 255
booting and installing, 255
compared to other installation methods, 34
description, 210
elements, 267, 275
booting and installing, 275
check script, 273
eng_profile creation, 271
engineering systems setup, 273
JumpStart directory, 271
marketing_profile creation, 272
marketing systems setup, 270, 275
networked, 209

Index 497
examples (Continued)
  non-networked, 206
  rules file editing, 272, 273
  site setup, 267, 268
  standalone system, 206
optional features, 235
  begin scripts, 235, 237
  finish scripts, 237, 241
  overview, 235
  site-specific installation programs, 248
overview, 209
preparing, 210, 234
Solaris Flash profile example, 229
tip line connection and, 258, 262
when upgrading, 71
custom_probes file
  naming, 250
  requirements, 250
  testing custom_probes, 254
  validating by using check, 253, 254
custom_probes.ok file
  creating, 253
  description, 253
Customizing, archives with scripts, 164
customizing, Solaris Flash archives, 160

D
  -d option, add_install_client command, 121
daemons, 448, 449
date and time, preconfiguring, 56
dd command, 219
defaults
  derived profile name, 237
  partitioning, 305
  designating disks, 308
  excluding disks, 294
  SI_CONFIG_DIR variable, 238
  software group installed, 293
deleting
  a Live Upgrade boot environment, 397
  clusters when upgrading, 293
  packages from software groups, 305
derived profiles, 236, 237
Developer Solaris Software Group, 292
description, 40
profile example, 226

Developer Solaris Software Group (Continued)
  size, 41
dfstab file, 214, 271
DHCP (Dynamic Host Configuration Protocol),
  preconfiguring, 56
differential archive
  creating, 180
  description, 157
  planning, 163
directories
  changing
    to image of Solaris 9 software on local
disk, 214
    to image of Solaris 9 SPARC Platform Edition software on local disk, 218
    to image of Solaris 9 x86 Platform Edition software on local disk, 220
    to JumpStart directory, 234, 253
JumpStart
  adding files, 238
  allowing access, 215
  copying files, 238
  copying installation files, 214, 218, 221
  creating directory, 271
  creating for systems, 217
  permissions, 213, 217
  rules file example, 222
  sharing directory, 214, 271
disk configuration files
  creating
    SPARC based systems, 243
    x86 based systems, 245
disk space
  planning, 39
  requirements for software groups, 41
diskettes
  copying Solaris 9 Device Configuration Assistant x86 Platform Edition
  diskette, 219
  formatting, 217, 220, 476
JumpStart directory
  creating for x86 based systems, 217
  JumpStart directory access, 215
diskless clients
  platforms, 291
  swap space, 291
disks probe keyword, description and values, 311
disksize rule keyword
description and values, 278, 311
display
tip line connection and custom JumpStart installation, 258, 262
display resolution, preconfiguring, 57
displaying
mounted file systems, 123
name of Live Upgrade boot environment, 398
platform name, 124
shared file systems, 124
system information, 124
domain name, preconfiguring, 56
domainname probe keyword, 311
domainname rule keyword, 278, 311
domains
probe keyword, 311
rule keyword, 278, 311
dontuse profile keyword, 294, 308

E
End User Solaris Software Group, 292
description, 40
size, 41
eng_profile example, 271
Entire Solaris Software Group, 292
description, 40
Entire Solaris software group, size, 41
Entire Solaris Software Group Plus OEM Support, 292
description, 40
size, 41
equal sign (=) in profile field, 236
/etc/bootparams file
enabling JumpStart directory access, 216, 448
/etc/dfs/dfstab file, 214, 271
/etc/locale file, 64
/etc/mnttab file, 217
exclamation mark (!) rule field, 223
existing partitioning value, 306
explicit partitioning value, 306
/export file system, 40

F
factory JumpStart, compared to other installation methods, 36
failed upgrade
rebooting problems, 453
recovery with Solaris Live Upgrade, 382
fdformat command, 217, 220, 476
fdisk command, 245
fdisk partition, requirements, 28
fdisk profile keyword, description and values, 294
fdisk profile keyword, example, 226
files and file systems
begin script output, 236
copying
a shareable file system for Solaris Live Upgrade, 356
JumpStart directory files using finish scripts, 238
JumpStart installation files, 214, 218, 221
Solaris 9 Device Configuration Assistant x86 Platform Edition diskette, 219
creating local file systems, 297
customizing for Live Upgrade, 343
displaying mounted file systems, 123
displaying shared file systems, 124
finish script output, 237
mounting remote file systems, 296
overview for Live Upgrade, 316, 317
sharing file systems in Live Upgrade, overview, 335
UFS creation, 218
filesys keyword, 297
filesys profile keyword
description and values, 296
examples, 226
finish.log file, 237
finish rule field, description, 223
finish scripts
adding files, 238
adding packages and patches, 238
customizing the root environment, 239
rule field, 223
setting the system’s root password, 240
flar create command, 198
Flash
See Solaris Flash
formatting diskettes, 217

Index 499
G
geo keyword, 299
getfile: RPC failed: error 5: RPC Timed out message, 216
graphics card, preconfiguring, 57

H
hard disks
fdisk partition requirements, 28
mounting, 296
partitioning
designating for partitioning default, 308 examples, 226
excluding for partitioning default, 294
profile keyword, 305
rootdisk values, 307
size
probe keywords, 311 312
root space, 291
rule keywords, 278 281 311 312
space available, 84, 87, 98, 103, 108, 114, 119
surface analysis for IDE drives, 450
swap space
allocating swap, 41
diskless client, 291
maximum size, 292
profile examples, 210, 226
size requirement for Solaris Web Start, 28
host name, preconfiguring, 56
hostaddress probe keyword, 311
hostaddress rule keyword, 278, 311
hostname probe keyword, description and values, 311
hostname rule keyword
description and values, 278, 311
example, 277

I
IDE interface
mapping out bad blocks, 450
surface analysis, 450
install_config command, 216
install server
creating with CD media, 97, 102
creating with CD media, example, 100, 105
creating with cross-platform CD media, 110 116
creating with cross-platform CD media, example, 111, 116
creating with cross-platform media, 106
creating with DVD, 86
creating with DVD, example, 85, 88
creating with DVD media, 83
creating with DVD media, example, 85
network installation setup with CD media, 120
network installation setup with DVD media, 91
on subnet, 85, 88, 100, 105, 111, 116
system types applicable, 77
install_type keyword, 300
install_type profile keyword examples, 226
requirement, 225, 226
testing profiles, 231, 232
installation
compared to upgrade, 25
comparison of methods, 33
disk space recommendations, 39
overview of tasks, 23
Solaris Flash archives, description, 155
system requirements, 27
installed probe keyword, description and values, 311
installed rule keyword
description and values, 278, 311
installing clone systems
initial installation, 155
updating, 157
IP addresses
preconfiguring, 56
preconfiguring a default route, 56
probe keyword, 311
rule keyword, 278, 311
specifying a default route, 46, 52
IPv6, preconfiguring, 56
IRQ level, preconfiguring, 57
isa_bits keyword, 301
J
JumpStart directory
adding files with finish scripts, 238
copying files
installation files, 214, 218, 221
using finish scripts, 238
creating
diskette for x86 based systems, 217, 219, 476
diskette for SPARC based systems, 217
example, 271
server, 213
permissions, 213, 217
rules file example, 222
sharing, 213, 271

K
karch probe keyword, 312
karch rule keyword, 279, 312
Kerberos
information to configure, 45
preconfiguring, 56
keyboard language and layout,
preconfiguring, 57
keywords
custom JumpStart
Solaris Flash archives, 283
probe, 249
Solaris Flash archives, 193
sysidcfg file, 59

L
layout_constraint keyword, 72, 301
LBA
See Logical Block Addressing
gle0: No carrier - transceiver cable problem
message, 443
Live Upgrade
See Solaris Live Upgrade
locale file, 64
locale keyword, 303
locale.org_dir table, adding entries, 65
log files
begin script output, 236
log files (Continued)
finish script output, 237
upgrade installation, 130, 138, 144, 150, 151
logical AND rule field, 223
Logical Block Addressing, 29

M
Makefile file, 63
mapping out bad blocks on IDE drives, 450
marketing_profile example, 272
master system
See also archive
customizing an installation of, 160
description, 160
how to install, 171
peripheral devices, 161
matching
derived profiles, 236
order for rules, 224, 255, 260
rootdisk values, 307
memory
probe keyword, 312
rule keyword, 279, 312
swap space size and, 292
memsize probe keyword, description and values, 312
memsize rule keyword
description and values, 279, 312
merging a Solaris Flash archive, 187
microprocessors
probe keywords, 311
rule keywords, 277, 311
mnttab file, 217
model probe keyword, description and values, 312
model rule keyword
description and values, 280, 312
modify_install_server, description, 123
monitor type, preconfiguring, 57
mount command, 123
mounting
begin script caution, 236
displaying mounted file systems, 123
by Solaris 9 installation, 237
remote file systems, 296
multiple lines in rules files, 222
N
name server, preconfiguring, 56
name service, preconfiguring, 56
names/naming
  custom probes file, 250
derived profile names, 237
host name, 121, 278 311
rules file, 222
software groups, 293
system model names, 280 312
system platform name determination, 124
netmask, preconfiguring, 56
network installation
custom JumpStart installation
  example, 209
description, 77
preparing, 77
requirements, 77
  using CD media, 97 101 118
  using cross-platform media, 106
  using DHCP, 79
  using DVD media, 82 89
  using PXE, 31 477
network interface, preconfiguring, 56
network number, 280 312
network probe keyword, description and values, 312
network rule keyword
  description and values, 280 312
nlistbladm command, 65 66
No carrier - transceiver cable problem
  message, 443
no_master_check keyword, 304
Not a UFS filesystem message, 443
num_clients profile keyword, 304

O
organization
Solaris 9 Software SPARC Platform Edition
  CDs, 421
Solaris 9 Software x86 Platform Edition
  CDs, 427
Solaris 9 SPARC Platform Edition DVD, 419
Solaris 9 SPARC Platform Edition Installation
  CD, 420
organization (Continued)
  Solaris 9 SPARC Platform Edition Languages
    CD, 422
  Solaris 9 x86 Platform Edition DVD, 425
  Solaris 9 x86 Platform Edition Installation
    CD, 427
  Solaris 9 x86 Platform Edition Languages
    CD, 429
  osname probe keyword, 312
  osname rule keyword, 280 312
output files
  begin script log, 236
  finish script log, 237
  upgrade log, 130 138 144 150 151
P
-p option of check script, 234 253
package profile keyword, description and values, 305
packages
  adding with a finish script, 238
  adding with chroot, 239
  adding with Live Upgrade, overview, 364
administration file, 235
requirements
  for an inactive boot environment for
    Solaris Live Upgrade, 464 466
  when using the JumpStart program and
    Solaris Live Upgrade, 463
partitioning
  examples, 226
  excluding disks, 294
  fdisk partitions, 226
  fdisk partitions, 294
  profile keyword, 305 308
  root slice location, 444
partitioning keyword, 305
password, root, 240 241
Patch Analyzer, 471
patches, 49
  adding with a finish script, 238
  adding with chroot, 239
  adding with Live Upgrade, overview, 364
paths
  check script, 234 253
permissions
begin scripts, 236
finish scripts, 237
JumpStart directory, 213, 217
pfinstall command, 71, 229
planning
comparison of installation methods, 33
disk space, 39
for a Solaris Flash installation, 160
for Solaris Live Upgrade, 329
initial installation compared to upgrade, 24
installing over the network, 29
overview of tasks, 23
system requirements, 27
system requirements for Live Upgrade, 329
platforms
diskless client, 291
install server setup, 121
matching system attributes and profiles, 224, 255, 260
name determination, 124
probe keywords, 312
rule keywords, 279, 312
system model names, 280, 312
pointing device, preconfiguring, 57
Power Management, 57, 66
Preboot Execution Environment (PXE), 31, 477
preconfiguring system configuration information
advantages, 55
choosing a method, 56
locale using NIS, 63
locale using NIS+, 65
Power Management, 66
using a name service, 57, 63
using sysidcfg file, 57
precreation script, creating, 172
predeployment script
creating, 173
description, 192
preparing for installation
custom JumpStart installation, 210, 234
information you need before installing, 44
information you need before upgrading, 50
preconfiguring system information advantages, 55
methods, 56
preparing for installation (Continued)
preparing the system for install, 43
preparing the system for upgrade, 49
probe keywords
arch, 311
disks, 311
domainname, 311
hostaddress, 311
hostname, 311
installed, 311
karch, 312
memsize, 312
model, 312
network, 312
osname, 312
rootdisk, 312
totaldisk, 312
probe rule keyword, description and values, 281
processors
probe keywords, 311
rule keywords, 277, 311
profile keywords, 282, 308
archive_location, 283
backup_media, 288
boot_device, 290
case sensitivity, 282
client_arch, 291
client_root, 291
client_swap, 291
counter
clusterr

dontuse
description and values, 294
examples, 226
usedisk and, 308
fdisk
description and values, 294
example, 226
filesys
description and values, 296
examples, 226
local file systems, 297
remote file systems, 296
forced_deployment
description and values, 299
geo
description and values, 299
profile keywords (Continued)

install_type
   description and values, 300
   examples, 226
   requirement, 225, 226
isa_bits
   description and values, 301
layout_constraint
   description and values, 301
local_customization
   description and values, 303
locale
   description and values, 303
no_master_check
   description and values, 304
num_clients, 304
package
   description and values, 305
partitioning
   description and values, 305
designating disks, 308
   examples, 226
   excluding disks, 294
root_device, 306
system_type
   description and values, 308
   examples, 226
usedisk
   description and values, 308
profiles
   comments in, 225
   creating, 225
derived profiles, 236, 237
description, 225
   examples, 226
   eng_profile, 271
   marketing_profile, 272
   Solaris Flash, 228, 229
matching systems to, 224, 255, 260
naming, 226, 368
   requirements, 222, 225
rule field, 223
testing, 71, 231, 232
prtvtoc command
   SPARC: creating disk configuration file, 243
   x86: disk configuration file creation, 245
PXE (Preboot Execution Environment), 31, 477

\textbf{R}

- \texttt{r} option of check script, 234, 254
reboot script, creating, 175
release of Solaris 9 software
   installed probe keyword, 311
   installed rule keyword, 279, 311
osname probe keyword, 312
osname rule keyword, 280, 312
remote file systems, mounting, 296
requirements
   custom_probes file, 250
disk space, 39
   fdisk partition, 28
memory, 27
network installation
   servers, 77
profiles, 222, 225
swap space, 28
to use Live Upgrade, 329
to use Solaris Web Start, 27, 29
reset command, 124
resetting display and terminal after I/O
   interrupts, 124
root (/) file systems, package requirements for
   an inactive boot environment, 464
root (/) file systems, profile example, 210
root_device keyword, 306
root environment, customizing with a finish
   script, 239
root password, preconfiguring, 56
root password, setting with a finish script, 240
root slice location, 444
rootdisk
   definition, 307
   slice value for filesys, 297
   value set by JumpStart, 307
RPC failed: error 5: RPC Timed out
   message, 216
RPC Timed out message, 216, 448
rule_keyword rule field, 223
rule keywords, 277
any
   description and values, 277, 311
arch, 277, 311
disksize
   description and values, 278, 311
domainname, 278, 311
hostaddress, 278, 311
rule keywords (Continued)
  hostname, 277, 278, 311
  installed
    description and values, 279, 311
  karch, 279, 312
  memsize, 279, 312
  model, 280, 312
  network, 280, 312
  osname, 280, 312
  probe, 281
  totaldisk, 281, 312
rule_value rule field, 223
rules
  derived profiles, 236, 237
  examples, 224
  field descriptions, 223
  matching order, 224, 255, 260
  multiple line rules, 222
  rootdisk matching rules, 307
  syntax, 223
  testing validity, 234, 254
rules file
  adding rules, 222
  comments in, 222
  creating, 222
  custom JumpStart example, 272, 273
  description, 222
  example, 222
  multiple line rules, 222
  naming, 222
  syntax, 223
  testing rules, 234
  validating by using check, 234
  custom JumpStart example, 273
  derived profiles and, 237
rules.ok file
  creating, 233
  description, 233
  matching order for rules, 224, 255, 260
scripts (Continued)
  finish scripts, 237, 241, 248
  Solaris Flash archives
    creating, 171, 174
    customizing, 164
    guidelines, 165
  predeployment, creating, 173
  reboot, 175
security
  root password, 240, 241
secuirty policy, preconfiguring, 56
servers
  JumpStart directory creation, 213
  network installation setup with CD media
    standalone installation, 119
  network installation setup with DVD media
    standalone installation, 91
  requirements for network installation, 77
  root space, 291
Service partition, preserving during installation
  (x86 based systems), 30
set_root_pw finish script, 240, 241
setup_install_server, description, 123
share command
  sharing JumpStart directory, 214, 271
shareable file systems, definition, 316
shareall command, 214, 271
sharing JumpStart directory, 214, 271
showmount command, 124
SI_CONFIG_DIR variable, 238
SI_PROFILE environment variable, 237
site-specific installation programs, 248
size
  fdisk partition, 295
  hard disk
    probe keywords, 311, 312
    root space, 291
    rule keywords, 278, 281, 311, 312
    space available, 84, 87, 98, 103, 108, 114, 119
  local file system, 297
  memory, 279, 312
  swap space
    diskless client, 291
    maximum size, 292
    minimum size for Solaris Web Start, 28
    profile examples, 210
    tip window dimensions, 258, 262

S
-s option of add_install_client command, 274
screen size, preconfiguring, 57
scripts
  begin scripts, 235, 237, 248
  Bourne shell scripts in rule fields, 223
slices
customizing file systems for Live Upgrade, 344
data files values, 297
guidelines for Live Upgrade, 333
probe keyword, 311
profile examples, 226
root slice location, 444
rule keyword, 279, 311
software groups
descriptions, 41
for profiles, 292, 293
profile examples, 226
sizes, 41
specifying packages, 305
upgrading, 27, 71, 293
Solaris 9 Device Configuration Assistant x86
Platform Edition diskette, 219
accessing, 31
Solaris 9 software
groups, 292, 293
profile examples, 226
specifying packages, 305
upgrading, 293
release or version
installed probe keyword, 311
installed rule keyword, 279, 311
Solaris Flash
description, 155
installing
on Live Upgrade boot environment, 372
Solaris suninstall program, SPARC, 143
Solaris suninstall program, x86, 150
Solaris Web Start, SPARC, 129
Solaris Web Start, x86, 137
keywords
custom JumpStart, 283
Solaris Live Upgrade
activating a boot environment, 378
cancel a job, 394
changing name of a boot environment, 399
commands, 413
compare boot environments, 395
configuring file systems, 343
copying
a shareable file system, 356
Solaris Live Upgrade (Continued)
creating
a boot environment, 341
creating a boot environment, 316
deleting a boot environment, 397
displaying name of boot environment, 398
failed upgrade recovery, 382
file system slices, 344
installing
a Solaris Flash archive, 372
a Solaris Flash archive with a profile, 376, 377
packages, 339
overview, 316
printing to a file, 344
starting, 340
stopping, 340
upgrade task map, 362
upgrading a boot environment, 362
viewing configuration of boot environments, 403
Solaris software
release or version
osname probe keyword, 312
osname rule keyword, 280, 312
Solaris suninstall program
compared to other installation methods, 34
ways to run, 141
Solaris Web Start
command-line interface (CLI), 125
compared to other installation methods, 33
graphical user interface (GUI), 125
installation program, 125
requirements
LBA on x86 systems, 29
swap size, 28
ways to run, 125
splitting a Solaris Flash archive, 186
standalone systems
custom JumpStart installation example, 206
profile examples, 226
starting
check script, 233, 234
rpld daemon, 449
tftpd daemon, 448, 449
status, display boot environment, 391
stty command, 43, 49, 258, 262
subnet
  boot server creation on with CD media, 117
  boot server creation on with DVD media, 89
  install server, 88, 110, 116
SUNWCall group, 292
SUNWCprog group, 292
SUNWCreq group, 292
SUNWCuser group, 292
SUNWCXall group, 292
surface analysis for IDE drives, 450
swap file systems
  allocating swap, 41
  customizing
    for Live Upgrade (character interface), 346
    for Live Upgrade (command line), 352
diskless client swap space, 291
memory size and, 292
overview for Solaris Live Upgrade, 316
profile examples, 210
size determination, 292
sysidcfg file
  guidelines and requirements, 57
  how to create, 61
  keywords, 59
  syntax, 58
system information, displaying, 124
system_type profile keyword
  description and values, 308
  examples, 226

T
terminal type, preconfiguring, 56
testing
  profiles, 71, 229, 231, 232, 370
  validating custom probes files
    testing custom probes, 254
    using check, 253
  validating rules file
    using check, 254
validating rules files
  custom JumpStart example, 273
  derived profiles and, 237
  testing rules, 234
  using check, 233, 234
tftpd daemon, 448, 449
time and date, preconfiguring, 56
time zone, preconfiguring, 56
timed out RPC error, 448
tip line connection and custom JumpStart installation, 258, 262
token ring card, booting error with, 447
totaldisk probe keyword, 312
totaldisk rule keyword, 281, 312
transceiver cable problem message, 443
troubleshooting
  booting from wrong server, 448
  general installation problems, 444
  booting the system, 448

U
UFS, 218
uname command, 124
Unknown client error message, 441
upgrading a clone system, description, 157
upgrade
  cleanup, 139
  compared to initial installation, 25, 71
  comparison of methods, 33
  disk space recommendations, 39
  methods, 26, 71
  overview of tasks, 23
  system requirements, 27
  to a Solaris Update release, 471
  with custom JumpStart, 71
  with Disk Space Reallocation, 72
  with Live Upgrade, overview, 321
  with Solaris Live Upgrade, 362
upgrade installation
  custom JumpStart installation, 255
  failed upgrade, 453
  failed upgrade recovery for Solaris Live Upgrade, 382
  log file, 130, 138, 144, 150, 151
  profile keywords, 293, 300, 305
  upgrade_log file, 130, 138, 144, 150, 151
  upgrading, cleanup, 151
  usedisk profile keyword, description and values, 308
user–defined section
  Solaris Flash archives
    creating, 173
validating
custom_probes file
testing, 254
using check, 254
profiles, 231
rules files
custom JumpStart example, 273
derived profiles and, 237
testing rules, 234
using check, 233, 234, 254
/var/sadm/system/logs/begin.log
file, 236
/var/sadm/system/logs/finish.log
file, 237
/var/sadm/install_data/upgrade_log file, 144, 150, 151
/var/yp/make command, 65
/var/yp/Makefile, 63
variables
SI_CONFIG_DIR, 238
SI_PROFILE, 237
SYS_MEMSIZE, 231
version of Solaris 9 software, installed rule keyword, 279
version of Solaris software
installed probe keyword, 311
installed rule keyword, 311
osname probe keyword, 312
osname rule keyword, 280, 312
volcheck command, 220
Volume Manager
copying, 217, 220
Solaris 9 Device Configuration Assistant image, 220
Solaris 9 Device Configuration Assistant x86 Platform Edition diskette, 219

WARNING: CHANGE DEFAULT BOOT DEVICE, 450

WARNING: clock gained xxx days message, 443
web proxy, preconfiguring, 57
wrapping lines in rules files, 222