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

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing. If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

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

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

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

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

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

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

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

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

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

Preface ..................................................................................................................................................... 7

1 Where to Find Oracle Solaris Installation Planning Information .................................................. 11
   Where to Find Planning and System Requirement Information .................................................. 11

2 JumpStart (Overview) ....................................................................................................................... 13
   JumpStart Introduction .................................................................................................................. 13
   JumpStart Example Scenario ...................................................................................................... 14
   How the JumpStart Program Installs Oracle Solaris Software ................................................... 15

3 Preparing JumpStart Installations (Tasks) .......................................................................................... 19
   Task Map: Preparing JumpStart Installations .............................................................................. 19
   Creating a Profile Server for Networked Systems ................................................................. 21
   ▼ How to Create a JumpStart Directory on a Server ................................................................. 21
   Allowing All Systems Access to the Profile Server .................................................................. 23
   Creating a Profile Diskette for Stand-alone Systems ............................................................... 24
   ▼ SPARC: How to Create a Profile Diskette ............................................................................. 25
   ▼ x86: How to Create a Profile Diskette With GRUB .............................................................. 26
   Creating the rules File .................................................................................................................. 28
   Syntax of the rules File ................................................................................................................. 29
   ▼ How to Create a rules File ................................................................................................. 29
   rules File Example ..................................................................................................................... 31
   Creating a Profile ......................................................................................................................... 32
   Syntax of Profiles ....................................................................................................................... 32
   ▼ How to Create a Profile ........................................................................................................... 33
   Profile Examples ......................................................................................................................... 33
   Testing a Profile ......................................................................................................................... 44
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼ How to Create a Temporary Oracle Solaris Environment to Test a Profile</td>
<td>45</td>
</tr>
<tr>
<td>▼ How to Test a Profile</td>
<td>46</td>
</tr>
<tr>
<td>Profile Test Example</td>
<td>48</td>
</tr>
<tr>
<td>Validating the rules File</td>
<td>49</td>
</tr>
<tr>
<td>▼ How to Validate the rules File</td>
<td>49</td>
</tr>
<tr>
<td>▼ How to Create a Compressed Configuration File</td>
<td>58</td>
</tr>
<tr>
<td>▼ How to Create a Compressed Configuration File</td>
<td>59</td>
</tr>
<tr>
<td>Creating Disk Configuration Files</td>
<td>59</td>
</tr>
<tr>
<td>▼ SPARC: How to Create a Single-Disk Configuration File</td>
<td>60</td>
</tr>
<tr>
<td>▼ SPARC: How to Create a Multiple-Disk Configuration File</td>
<td>61</td>
</tr>
<tr>
<td>▼ x86: How to Create a Single-Disk Configuration File</td>
<td>62</td>
</tr>
<tr>
<td>▼ x86: How to Create a Multiple-Disk Configuration File</td>
<td>64</td>
</tr>
<tr>
<td>Creating Custom Rule and Probe Keywords (Tasks)</td>
<td>67</td>
</tr>
<tr>
<td>_probe Keywords</td>
<td>67</td>
</tr>
<tr>
<td>Creating a custom_probes File</td>
<td>68</td>
</tr>
<tr>
<td>Syntax of the custom_probes File</td>
<td>68</td>
</tr>
<tr>
<td>Syntax of Function Names in the custom_probes File</td>
<td>68</td>
</tr>
<tr>
<td>▼ How to Create a custom_probes File</td>
<td>69</td>
</tr>
<tr>
<td>Validating the custom_probes File</td>
<td>70</td>
</tr>
<tr>
<td>▼ How to Validate the custom_probes File</td>
<td>70</td>
</tr>
</tbody>
</table>
Preface

This book describes how to install and upgrade the Oracle Solaris operating system (OS) on both networked and nonnetworked SPARC and x86 architecture based systems. This book covers using the JumpStart, a feature of Oracle Solaris, installation method and the creation of RAID-1 volumes during installation.

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

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

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

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

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

Who Should Use This Book

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

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

Table P–1 lists documentation for system administrators.

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


Typographic Conventions

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

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

Shell Prompts in Command Examples

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

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bash shell, Korn shell, and Bourne shell</td>
<td><code>$</code></td>
</tr>
<tr>
<td>Shell</td>
<td>Prompt</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>Bash shell, Korn shell, and Bourne shell for superuser</td>
<td>#</td>
</tr>
<tr>
<td>C shell</td>
<td>machine_name%</td>
</tr>
<tr>
<td>C shell for superuser</td>
<td>machine_name#</td>
</tr>
</tbody>
</table>
This book provides information on how to use the automated JumpStart, a feature of Oracle Solaris, installation program to install the Oracle Solaris operating system. This book provides all you need to know about installing with the JumpStart program, but a planning book in our collection of installation documentation might be useful to read before you begin preparing for a JumpStart installation. The following references provide useful information before you install your system.

---

**Where to Find Planning and System Requirement Information**

The Oracle Solaris 10 8/11 Installation Guide: Planning For Installation and Upgrade provides system requirements and high-level planning information, such as planning guidelines for file systems, and upgrade planning and much more. This section provides an overview of the chapters for this book.

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

<table>
<thead>
<tr>
<th>Chapter Descriptions From the Planning Guide</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>These chapters provide overviews of several technologies that relate to Oracle Solaris OS installation or upgrade. Guidelines and requirements related to these technologies are also included. These chapters include information about ZFS installations, booting, Oracle Solaris Zones partitioning technology, and RAID-1 volumes that can be created at installation.</td>
<td>Part II, &quot;Understanding Installations Related to ZFS, Booting, Oracle Solaris Zones, and RAID-1 Volumes,&quot; in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</td>
</tr>
</tbody>
</table>
This chapter provides an introduction and overview to the JumpStart, a feature of Oracle Solaris, installation process.

Note – If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, “Installing a ZFS Root Pool With JumpStart,” for limitations and profile examples.

This chapter covers the following topics:
- “JumpStart Introduction” on page 13
- “How the JumpStart Program Installs Oracle Solaris Software” on page 15

JumpStart Introduction

The 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 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 JumpStart installation is completely automated.
Note – The root password for your client systems can be preset prior to the installation by including the root_password keyword in the sysidcfg. See the sysidcfg(4) man page.

Auto Registration, a feature of Oracle Solaris, is new in the Oracle Solaris 10 9/10 release. When you install or upgrade your system, configuration data about your system is, on rebooting, automatically communicated through the existing service tag technology to the Oracle Product Registration System. This service tag data about your system is used, for example, to help Oracle enhance customer support and services.

If you include the auto_reg keyword in the sysidcfg file prior to the installation or upgrade, the installation can remain completely automated. However, if you do not include the auto_reg keyword, you will be prompted to provide your support credentials and proxy information for Auto Registration during the installation or upgrade.

JumpStart Example Scenario

This chapter describes the JumpStart process by using an example scenario. In this example scenario, the systems are set up with the following parameters:

- Oracle Solaris needs to be installed on 100 new systems.
- Seventy of the systems are SPARC based systems that are owned by the engineering group and need to be installed as stand-alone systems with the Oracle Solaris OS software group for developers.
- The remaining 30 systems are x86 based, and are owned by the marketing group. They need to be installed as stand-alone systems with the Oracle Solaris OS 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 Oracle 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 Oracle 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 eng_profile profile which was created for the engineering group. In the rule for the marketing group, a link is added to the market_profile 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 JumpStart installations on nonnetworked, stand-alone systems.
- A profile server is used when you want to perform 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 file is a generated version of the rules file that the JumpStart program uses to install the Oracle Solaris software.

How the JumpStart Program Installs Oracle Solaris Software

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

The following figure illustrates how a JumpStart installation works on a stand-alone, nonnetworked system. The system administrator initiates the 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 Oracle Solaris software. The JumpStart program reads Pete’s profile and installs the Oracle Solaris software, based on the instructions that the system administrator specified in Pete’s profile.
The following figure illustrates how a JumpStart installation works with more than one system on a network. Previously, the system administrator set up different profiles and saved the profiles on a single server. The system administrator initiates the JumpStart installation on one of the engineering systems. The JumpStart program accesses the rules files in the JumpStart/ directory on the server. It matches the engineering system to rule 1. rule 1 specifies that the JumpStart program use Engineering Group’s Profile to install the Oracle Solaris software.
The JumpStart program reads Engineering Group’s Profile and installs the Oracle Solaris software, based on the instructions that the system administrator specified in Engineering Group’s Profile.

FIGURE 2–2  How a JumpStart Installation Works: Networked Example

The following figure describes the order in which the JumpStart program searches for JumpStart files.
The system proceeds with a custom JumpStart installation. The profile specified in the matched rule is used to install the system.
Prepating 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 Oracle Solaris software by using the JumpStart, installation method.

**Note** – If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, "Installing a ZFS Root Pool With JumpStart," for limitations and profile examples.

This chapter covers the following topics:

- "Task Map: Preparing JumpStart Installations" on page 19
- "Creating a Profile Server for Networked Systems" on page 21
- "Creating a Profile Diskette for Stand-alone Systems" on page 24
- "Creating the rules File" on page 28
- "Creating a Profile" on page 32
- "Testing a Profile" on page 44
- "Validating the rules File" on page 49

**Task Map: Preparing 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 Oracle Solaris software is installed on the system.</td>
<td>If a previous release of Oracle 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>“Upgrade Planning” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</td>
</tr>
</tbody>
</table>
### TABLE 3–1  Task Map: Preparing JumpStart Installations  
(Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
</table>
| Create a JumpStart directory.             | **On a server**<br>If you want to perform 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 JumpStart files.  
**On a diskette**<br>If you want to perform JumpStart installations on systems that are not connected to a network, you must create a profile diskette. A profile diskette contains the JumpStart files. | “Creating a Profile Server for Networked Systems” on page 21  
“Creating a Profile Diskette for Stand-alone Systems” on page 24 |
| 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 28 |
| Create a profile for every rule.          | A profile is a text file that defines how to install the Oracle 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 Oracle 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 32 |
| (Optional) Test the profiles.             | After you create a profile, use the `ptinstall(1M)` command to test the profile before you use the profile to install or upgrade a system.                                                                     | “Testing a Profile” on page 44 |
| Validate the rules file.                  | The `rules.ok` file is a generated version of the `rules` file that the JumpStart program uses to match the system to be installed with a profile. You must use the check script to validate the `rules` file.                                | “Validating the rules File” on page 49 |
Creating a Profile Server for Networked Systems

When setting up JumpStart installations for systems on the network, you need to create a JumpStart directory on a server. The JumpStart directory contains all of the essential 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 JumpStart files for different platforms. For example, an x86 server can provide JumpStart files for both SPARC based systems and x86 based systems.

Note – After you create a profile server, you must allow systems to access the server. For detailed instructions, see “Using a Wildcard Entry to Allow All Systems Access to the Profile Server” on page 23.

▼ How to Create a JumpStart Directory on a Server

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

1 Locate the server on which you want to create the JumpStart directory.

2 Become superuser or assume an equivalent role.

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

3 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
```

4 Add the following entry to the `/etc/dfs/dfstab` file.

```bash
share -F nfs -o ro,anon=0 jumpstart-dir-path
```

Chapter 3 • Preparing JumpStart Installations (Tasks) 21
For example, the following entry shares the /jumpstart directory:

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

5 Type `shareall` and press Return.

6 (Optional) Copy examples of JumpStart files to your JumpStart directory.

   a. Access the installation disc or image.

   b. Copy the example JumpStart files into the JumpStart directory on the profile server.

   ```
   # cp -r media-path/Solaris_10/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
                       JumpStart files
   
   For example, the following command copies the jumpstart_sample directory into the
   /jumpstart directory on the profile server:

   cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /jumpstart
   
   c. Update the example JumpStart files so that the files work in your environment.

7 Ensure that root owns the JumpStart directory and that the permissions are set to 755.

8 Allow systems on the network to access the profile server.

   For detailed instructions, see “Using a Wildcard Entry to Allow All Systems Access to the
   Profile Server” on page 23.
Allowing 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 JumpStart installation. Use one of the methods described in the following table to ensure access.

<table>
<thead>
<tr>
<th>Command or File</th>
<th>Providing Access</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_install_client command</td>
<td>Each time that you add a system for network installation, use the -c option with the add_install_client command to specify the profile server.</td>
<td>▪ For DVD media, see &quot;Adding Systems to Be Installed From the Network With a DVD Image&quot; in Oracle Solaris 10/13 Installation Guide: Network-Based Installations</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> – If you are not using NFS, then you must use another means to provide access.</td>
<td>▪ For CD media, see &quot;Adding Systems to Be Installed From the Network With a CD Image&quot; in Oracle Solaris 10/13 Installation Guide: Network-Based Installations</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>For SPARC based systems</strong>, use the boot command.</td>
<td>▪ &quot;Creating a Compressed Configuration File&quot; on page 58</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>For x86 based systems</strong>, edit the GRUB menu.</td>
<td>▪ Step 3 in &quot;SPARC: To Perform an Installation or Upgrade With the JumpStart Program&quot; on page 78</td>
</tr>
<tr>
<td></td>
<td>GRUB is a feature of Oracle Solaris.</td>
<td>▪ &quot;x86: How to Perform an Installation or Upgrade With the JumpStart Program and GRUB&quot; on page 82</td>
</tr>
</tbody>
</table>

Specify the location of the JumpStart directory when you boot the system

|                     | **For SPARC based systems**, use the boot command to boot the system. Specify the location of the JumpStart directory on the profile server when you boot the system. You must compress the JumpStart configuration files into one file. Then, save the compressed configuration file on an HTTP or HTTPS server. | ▪ "Creating a Compressed Configuration File" on page 58                     |
|                     | **For x86 based systems**, specify the location of the JumpStart directory on the profile server when you boot the system by editing the boot entry on the GRUB menu. You must compress the JumpStart configuration files into one file. Then, save the compressed configuration file on an HTTP or HTTPS server. When you edit the GRUB menu entry, specify the location of the compressed file. | ▪ "x86: How to Perform an Installation or Upgrade With the JumpStart Program and GRUB" on page 82 |

/etc/bootparams file or naming service bootparam database

Add a wildcard.

Using a Wildcard Entry to Allow All Systems Access to the Profile Server

You can use a wildcard entry to allow all systems access to the profile server only if you store network installation information in the following places:
In the /etc/bootsparams file
In the naming service bootsparams database

The systems must be of the same type, such as all SPARC systems.

Add the following entry to the file or database:

* install_config=server:jumpstart-dir-path
* A 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 enables 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 installation client is booted:

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

"Booting From the Network, Error Messages" in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations contains details about this error message.

All systems can now access the profile server.

### Creating a Profile Diskette for Stand-alone 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 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.

See one of the following procedures:

- "SPARC: How to Create a Profile Diskette" on page 25
“x86: How to Create a Profile Diskette With GRUB” on page 26

\section*{SPARC: How to Create a Profile Diskette}

\textbf{Note} – This procedure assumes that the system is running Solaris Volume Manager. If you are not using Solaris Volume Manager to manage diskettes, CDs, and DVDs, refer to \textit{System Administration Guide: Devices and File Systems} for detailed information about managing removable media without Solaris Volume Manager.

1 \textbf{Locate a SPARC based system to which a diskette drive is attached.}

2 \textbf{Become superuser or assume an equivalent role.}

Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in \textit{System Administration Guide: Security Services}.

3 \textbf{Insert a blank diskette or a diskettethat can be overwritten in the diskette drive.}

4 \textbf{Mount the diskette.}

\begin{verbatim}
# volcheck
\end{verbatim}

5 \textbf{Determine whether the diskette contains a UNIX file system (UFS) by examining the contents of the file /etc/mnttab on the system for an entry such as the following:}

\begin{verbatim}
/vol/dev/diskette0/scrap /floppy/scrap ufs suid,rw,largefiles,dev=1740008 927147040
\end{verbatim}

\begin{itemize}
  \item If the entry exists, go to the next step.
  \item If the entry does not exist, create a UFS on the diskette.
\end{itemize}

\begin{verbatim}
# newfs /vol/dev/aliases/floppy0
\end{verbatim}

6 \textbf{(Optional) Copy examples of JumpStart files to your JumpStart directory.}

a. \textbf{Access the installation disc or image.}

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Example Locations} & \textbf{Instructions} \\
\hline
The Oracle Solaris Operating System for SPARC Platforms DVD or the Oracle Solaris Software for SPARC Platforms - 1 CD & Insert the Oracle Solaris Operating System for SPARC Platforms DVD or the Oracle Solaris Software for SPARC Platforms - 1 CD into the server’s CD-ROM drive. \\
& Solaris Volume Manager automatically mounts the CD or DVD. \\
\hline
\end{tabular}
\end{table}
Creating a Profile Diskette for Stand-alone Systems

<table>
<thead>
<tr>
<th>Example Locations</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>An image of the Oracle Solaris Operating System for SPARC Platforms DVD or the Oracle Solaris Software for SPARC Platforms - 1 CD on a local disk</td>
<td>Change the directory to the location of the Oracle Solaris Operating System for SPARC Platforms DVD or the Oracle Solaris Software for SPARC Platforms - 1 CD image. For example, type the following command: <code>cd /export/install</code></td>
</tr>
</tbody>
</table>

b. **Copy the example JumpStart files into the JumpStart directory on the profile diskette.**

```
# cp -r media-path/Solaris_10/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 JumpStart files

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

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

```
cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

c. **Update the example JumpStart files on the profile diskette so that the files work in your environment.**

7 **Ensure that root owns the JumpStart directory and that permissions are set to 755.**

8 **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 JumpStart installations. To continue, go to “Creating the rules File” on page 28.

▼ **x86: How to Create a Profile Diskette With GRUB**

Use this procedure to create a profile diskette with GRUB. A GRUB menu is provided during the installation procedure that enables the boot process. The GRUB menu replaces the Oracle Solaris Device Configuration Assistant that might have been needed to boot a system in past releases.
Note – This procedure assumes that the system is running Solaris Volume Manager. If you are not using Solaris Volume Manager to manage diskettes, CDs, and DVDs, refer to System Administration Guide: Devices and File Systems for detailed information about managing removable media without Solaris Volume Manager.

1 Locate an x86 based system to which a diskette drive is attached.

2 Become superuser or assume an equivalent role.
   Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in System Administration Guide: Security Services.

3 Insert a blank diskette or a diskette that can be overwritten into the diskette drive.

4 Mount the diskette.
   # volcheck

5 (Optional) Copy examples of JumpStart files to your JumpStart directory.

   a. Access the installation disc or image.

      | Example Locations                                                                 | Instructions                                                                 |
      |----------------------------------------------------------------------------------|------------------------------------------------------------------------------|
      | The Oracle Solaris Operating System for x86 Platforms DVD or the Oracle Solaris Software for x86 Platforms - 1 CD | Insert the Oracle Solaris Operating System for x86 Platforms DVD or the Oracle Solaris Software for x86 Platforms - 1 CD into the server’s CD-ROM drive. Solaris Volume Manager automatically mounts the DVD or CD. |
      | An image of the Oracle Solaris Operating System for x86 Platforms DVD or the Oracle Solaris Software for x86 Platforms - 1 CD on a local disk | Change directory to the location of the Oracle Solaris Operating System for x86 Platforms DVD or the Oracle Solaris Software for x86 Platforms - 1 CD image. For example, type the following:
                                                                  | cd /export/install                                                                 |

   b. Copy the example JumpStart files into the JumpStart directory on the profile diskette.

      # cp -r media-path/Solaris_10/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 JumpStart files
Note – You must place all JumpStart installation files in the root (/) directory on the profile diskette.

For example, the following command copies the contents of jumpstart_sample on the Oracle Solaris Software for x86 Platforms - 1 CD to the root (/) directory on a profile diskette that is named scrap:

```
    cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

c. Update the example JumpStart files on the profile diskette so that the files work in your environment.

6 Ensure that root owns the JumpStart directory and that permissions are set to 755.

7 Eject the diskette by clicking Eject Disk in the File Manager window or by typing `eject floppy` on the command line.

8 In the Removable Media Manager dialog box, click OK.

9 Manually eject the diskette.

Next Steps 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 JumpStart installations. To continue, go to “Creating the rules File” on page 28.

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 Oracle Solaris OS. 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 Oracle 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 JumpStart installations.
Note – If you set up the JumpStart directory by using the procedures in “Creating a Profile Diskette for Stand-alone Systems” on page 24 or “Creating a Profile Server for Networked Systems” on page 21, an example rules file is already located in the JumpStart directory. The sample rules file contains documentation and some example rules. If you use the sample rules file, ensure that you comment out the example rules you do not intend to use.

Syntax of the rules File

The rules file must have the following attributes:

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

The rules file can contain any of the following:

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

How 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 Oracle Solaris software.

For a list of rules file keywords and values, see “Rule Keywords and Values” on page 97.

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

\! rule-keyword rule-value \&\& \! rule-keyword rule-value ... begin profile finish

\!

A symbol that is used before a keyword to indicate negation.

rule-keyword

A predefined lexical unit or word that describes a general system attribute, such as host name, hostname, or memory size, memsize. rule-keyword 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 97.
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 Oracle Solaris software that is specified by the profile that is listed in the profile field.

For a complete list of `rules` file limitations, see “Syntax of the `rules` File” on page 29.

**EXAMPLE 3–1  rule File**

```
# rule keywords and rule values begin script profile finish script
# ----------------------------- ------------ -------- -------------
hostname eng-1 - basic_prof -
network 192.168.255.255 && !model \ 'SUNW,Sun-Blade-100' - net_prof -
model SUNW,SPARCstation-LX - lx_prof complete
network 192.168.2.0 && karch i86pc setup x86_prof done
memsize 64-128 && arch i386 - prog_prof -
any - - generic_prof -
```

Some of the keywords and values in this example are as follows:

- **hostname** The rule matches if the system’s host name is `eng-1`. The `basic_prof` profile is used to install the Oracle Solaris software on the system that matches the rule.

- **network** The rule matches if the system is on subnet 192.168.255.255 and if the system is **not a Sun Blade 100** (`SUNW,Sun-Blade-100`). The `net_prof` profile is used to install the Oracle Solaris software on systems that match this rule. This rule also provides an example of continuing a single rule onto a new line by using the backslash character (`\`).

- **model** The rule matches if the system is a SPARCstation LX. The `lx_prof` profile and the `complete` finish script are used to install the Oracle Solaris software on systems that match this rule.

- **network** The rule matches if the system is on subnet 192.168.2.0 and is an x86 based system. The `setup` begin script, the `x86u_prof` profile, and the `done` finish script are used to install the Oracle Solaris software on systems that match the rule.

- **memsize** The rule matches if the system has between 64 and 128 MB of memory and is an x86 based system. The `prog_prof` profile is used to install the Oracle Solaris software on systems that match the rule.

- **any** The rule matches any system that did not match the previous rules. The `generic_prof` profile is used to install the Oracle 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 Oracle 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 Oracle 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** – Sample profiles are already located in the JumpStart directory if you created the JumpStart directory by using either of these procedures:

- “Creating a Profile Server for Networked Systems” on page 21
- “Creating a Profile Diskette for Stand-alone Systems” on page 24

## 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
How to Create a Profile

1. Use a text editor to create a text file or open a sample profile in the JumpStart directory that you created.
   Name the profile to reflect how you intend to use the profile to install the Oracle 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 101.
   
   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. (Optional) Test the profile.
   “Testing a Profile” on page 44 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 Oracle Solaris software is installed on a system. “Profile Keywords and Values” on page 101 contains a description of profile keywords and values.

Note – If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, “Installing a ZFS Root Pool With JumpStart,” for limitations and profile examples.

EXAMPLE 3–2  Mounting Remote File Systems and Adding and Deleting Packages

<table>
<thead>
<tr>
<th>profile keywords</th>
<th>profile values</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_type</td>
<td>initial_install</td>
</tr>
<tr>
<td>system_type</td>
<td>standalone</td>
</tr>
<tr>
<td>partitioning</td>
<td>default</td>
</tr>
<tr>
<td>filesys</td>
<td>any 512 swap</td>
</tr>
<tr>
<td>cluster</td>
<td>SUNWCprog</td>
</tr>
<tr>
<td>package</td>
<td>SUNWman delete</td>
</tr>
<tr>
<td>cluster</td>
<td>SUNWCacc</td>
</tr>
</tbody>
</table>

Some of the keywords and values in this example are as follows:


**EXAMPLE 3–2  Mounting Remote File Systems and Adding and Deleting Packages  (Continued)**

- **install_type**
  - The `install_type` keyword is required in every profile.

- **system_type**
  - The `system_type` keyword defines that the system is to be installed as a stand-alone system.

- **partitioning**
  - The file system slices are determined by the software to be installed with the value `default`. The size of `swap` is set to 512 MB and is installed on any disk, value `any`.

- **cluster**
  - The Developer Oracle Solaris Software Group, `SUNWCprog`, is installed on the system.

- **package**
  - 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 3–3  Mounting Remote File Systems and Adding a Third-Party Package**

```
# profile keywords    profile values
# ----------------- -------------------
install_type          initial_install
system_type           standalone
partitioning          default
filesys               any 512 swap   # specify size of /swap
cluster               SUNWCprog
cluster               SUNWCacc
package               apache_server /
                       http://package.central/packages/apache timeout 5
```

Some of the keywords and values in this example are as follows:

- **install_type**
  - The `install_type` keyword is required in every profile.

- **system_type**
  - The `system_type` keyword defines that the system is to be installed as a stand-alone system.

- **partitioning**
  - The file system slices are determined by the software to be installed with the value `default`. The size of `swap` is set to 512 MB and is installed on any disk, value `any`.

- **cluster**
  - The Developer Oracle Solaris Software Group, `SUNWCprog`, is installed on the system.

- **package**
  - A third-party package is installed on the system located on an HTTP server.

**EXAMPLE 3–4  Specifying Where to Install File Systems**

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

EXAMPLE 3–4 Specifying Where to Install File Systems  (Continued)

<table>
<thead>
<tr>
<th>keyword</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_type</td>
<td>initial_install</td>
</tr>
<tr>
<td>system_type</td>
<td>standalone</td>
</tr>
<tr>
<td>partitioning</td>
<td>explicit</td>
</tr>
<tr>
<td>filesys</td>
<td>c0t0d0s0 auto /</td>
</tr>
<tr>
<td>filesys</td>
<td>c0t3d0s1 auto swap</td>
</tr>
<tr>
<td>filesys</td>
<td>any auto usr</td>
</tr>
<tr>
<td>cluster</td>
<td>SUNWCall</td>
</tr>
</tbody>
</table>

Some of the keywords and values in this example are as follows:

**partitioning** 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.

**cluster** The Entire Oracle Solaris Software Group, SUNWCall, is installed on the system.

EXAMPLE 3–5 Upgrading and Installing Patches

<table>
<thead>
<tr>
<th># profile keywords</th>
<th>profile values</th>
</tr>
</thead>
<tbody>
<tr>
<td># -----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>install_type</td>
<td>upgrade</td>
</tr>
<tr>
<td>root_device</td>
<td>c0t3d0s2</td>
</tr>
<tr>
<td>backup_media</td>
<td>remote_filesystem timber:/export/scratch</td>
</tr>
<tr>
<td>package</td>
<td>SUNWbcp delete</td>
</tr>
<tr>
<td>package</td>
<td>SUNWxmlman add</td>
</tr>
<tr>
<td>cluster</td>
<td>SUNWCacc add</td>
</tr>
</tbody>
</table>
| patch              | patch_list nfs://patch_master/Solaris_10/patches \
|                   | retry 5       |
| locale             | de            |

Some of the keywords and values in this example are as follows:

**install_type** 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.

**root_device** The root file system on c0t3d0s2 is upgraded.

**backup_media** 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 109.

**package** The binary compatibility package, SUNWbcp, is not installed on the system after the upgrade.
package  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.

patch  A list of patches that are installed with the upgrade. The patch list is located on an NFS server named patch_master under the directories Solaris_10/patches. In case of a mount failure, the NFS mount is tried five times.

locale  The German localization packages are to be installed on the system.

EXAMPLE 3–6  Reallocating Disk Space for an Upgrade

```
# profile keywords profile values
# ---------------- -------------------
install_type upgrade
root_device c0t3d0s2
backup_media remote filesystem timber:/export/scratch
layout_constraint c0t3d0s2 changeable 100
layout_constraint c0t3d0s4 changeable
layout_constraint c0t3d0s5 movable
package SUNWbcp delete
package SUNWxwman add
cluster SUNWCacc add
locale de
```

Some of the keywords and values in this example are as follows:

**install_type**  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.

**root_device**  The root file system on c0t3d0s2 is upgraded.

**backup_media**  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 109.

**layout_constraint**  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.

**package**  The binary compatibility package, SUNWbcp, is not installed on the system after the upgrade.
package  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.

locale  The German localization packages are to be installed on the system.

EXAMPLE 3–7  Retrieving a Flash Archive From an HTTP Server

In the following example, the profile indicates that the JumpStart program retrieves the flash archive from an HTTP server. (Flash Archive is a feature of Oracle Solaris.)

```bash
# profile keywords     profile values
# ---------------- -------------------
install_type flash_install
archive_location http://192.168.255.255/flasharchive/solarisarchive
partitioning explicit
filesys c0t1d0s0 4000 /
filesys c0t1d0s1 512 swap
filesys c0t1d0s7 free /export/home
```

Some of the keywords and values in this example are as follows:

install_type  The profile installs a flash archive on the clone system. All files are overwritten as in an initial installation.

archive_location  The flash archive is retrieved from an HTTP server.

partitioning  The file system slices are determined by the filesys keywords, value explicit. The size of root (/) is based on the size of the flash archive. The root file system 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 remaining disk space. /export/home is installed on c0t1d0s7.

EXAMPLE 3–8  Retrieving a Flash Archive From a Secure HTTP Server

In the following example, the profile indicates that the JumpStart program retrieves the flash archive from a secure HTTP server.

```bash
# profile keywords     profile values
# ---------------- -------------------
install_type flash_install
archive_location https://192.168.255.255/solarisupdate.flar
partitioning explicit
filesys c0t1d0s0 4000 /
filesys c0t1d0s1 512 swap
filesys c0t1d0s7 free /export/home
```

Some of the keywords and values in this example are as follows:
install_type

archive_location

partitioning

install_type The profile installs a flash archive on the clone system. All files are overwritten as in an initial installation.

archive_location The compressed flash archive is retrieved from a secure HTTP server.

partitioning The file system slices are determined by the filesys keywords, value explicit. The size of root (/) is based on the size of the flash archive. The size of swap is set to the necessary size and is installed on c0t1d0s1. /export/home is based on the remaining disk space. /export/home is installed on c0t1d0s7.

package The SUNWnew package is added from the Solaris_10 directory from the HTTP server 192.168.254.255.

Some of the keywords and values in this example are as follows:

install_type The profile installs a flash archive on the clone system. All files are overwritten as in an initial installation.

archive_location The flash archive is retrieved from an HTTP server.

partitioning The file system slices are determined by the filesys keywords, value explicit. The size of root (/) is based on the size of the flash archive. The root file system 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 remaining disk space. /export/home is installed on c0t1d0s7.

package The SUNWnew package is added from the Solaris_10 directory from the HTTP server 192.168.254.255.

EXAMPLE 3–10 Retrieving a Flash Archive Differential Archive From an NFS Server

In the following example, the profile indicates that the JumpStart program retrieves the 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.
### EXAMPLE 3–10  Retrieving a Flash Archive Differential Archive From an NFS Server  
(Continued)

<table>
<thead>
<tr>
<th># profile keywords</th>
<th>profile values</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_type</td>
<td>flash_update</td>
</tr>
</tbody>
</table>
| archive_location    | nfs installserver:/export/solaris/flasharchive \ 
                    | /solarisdiffarchive |
| no_master_check     |                |

Some of the keywords and values in this example are as follows:

- **install_type**
  - The profile installs a flash archive differential archive on the clone system. Only files that are specified by the archive are installed.

- **archive_location**
  - The flash archive is retrieved from an NFS server.

- **no_master_check**
  - The clone system is not checked for a valid system image. A valid system image would have been built from the original master system.

### EXAMPLE 3–11  Creating an Empty Boot Environment

In the following example, the profile indicates that the JumpStart program creates an empty boot environment. An empty boot environment contains no file systems and no copy from the current boot environment occurs. The boot environment can be populated later with a flash archive and then activated.

<table>
<thead>
<tr>
<th># profile keywords</th>
<th>profile values</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_type</td>
<td>initial_install</td>
</tr>
<tr>
<td>system_type</td>
<td>standalone</td>
</tr>
<tr>
<td>partitioning</td>
<td>explicit</td>
</tr>
<tr>
<td>filesystem</td>
<td>c0t0d0s0 auto /</td>
</tr>
<tr>
<td>filesystem</td>
<td>c0t3d0s1 auto swap</td>
</tr>
<tr>
<td>filesystem</td>
<td>any auto usr</td>
</tr>
<tr>
<td>cluster</td>
<td>SUNWCall</td>
</tr>
</tbody>
</table>
| bootenv createbe    | bename second BE \ 
                    | filesystem /:/dev/dsk/c0t1d0s0:ufs \ 
                    | filesystem -:/dev/dsk/c0t1d0s0:swap \ 
                    | filesystem /export:shared:ufs |

Some of the keywords and values in this example are as follows:

- **partitioning**
  - The file system slices are determined by the `filesystem` 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`.

- **cluster**
  - The Entire Oracle Solaris Software Group, SUNWCall, is installed on the system.
Creating an Empty Boot Environment

(Continued)

bootenv createbe

An empty, inactive boot environment is set up on disk c0t1d0. File systems for root (/), swap, and /export are created, but left empty. This second boot environment can be installed with a flash archive at a later time. The new boot environment can then be activated to become the current boot environment.

For keyword values and background about using this keyword, see the following references:

- For descriptions of keyword values, see “Profile Keywords and Values” on page 101.
- For background about using a flash archive, see Chapter 1, “Flash Archive Overview,” in Oracle Solaris 10/13 Installation Guide: Flash Archives (Creation and Installation).

Creating RAID-1 Volumes When Installing a Flash Archive

In the following example, the profile indicates that the JumpStart program uses Solaris Volume Manager technology to create RAID-1 volumes (mirrors) for the root (/), swap, /usr and /export/home file systems. A flash archive is installed on the boot environment.

```bash
# profile keywords  profile values
# ----------------- -------------------
install_type        flash_install
archive_location    nfs server:/export/home/export/flash.s10.SUNWCall
partitioning        explicit
filesys             mirror:d10 c0t0d0s0 c0t1d0s0 4096 /
                   mirror:c0t0d0s1 2048 swap
filesys             mirror:d30 c0t0d0s3 c0t1d0s3 4096 /usr
filesys             mirror:d40 c0t0d0s4 c0t1d0s4 4096 /usr
filesys             mirror:d50 c0t0d0s5 c0t1d0s5 free /export/home
metadb              c0t1d0s7 size 8192 count 3
```

Some of the keywords and values in this example are as follows:

install_type The profile installs a flash archive on the clone system. All files are overwritten as in an initial installation.

archive_location The flash archive is retrieved from an NFS server.

partitioning The file system slices are determined by the filesys keywords, value explicit.
EXAMPLE 3–12 Creating RAID-1 Volumes When Installing a Flash Archive  (Continued)

filesys
The root (/) file system is created and mirrored on the slices c0t0d0s0 and c0t1d0s0. The size of the root (/) file system is set to 4096 MB. The RAID-1 volume that mirrors c0t0d0s0 and c0t1d0s0 is named d10.

filesys
The swap file system is created and mirrored on the slice c0t0d0s1, and is sized at 2048 MB. The JumpStart program assigns a name to the mirror.

filesys
The /usr file system is created and mirrored on the slices c0t1d0s3 and c0t0d0s3. The size of the /usr file system is set to 4096 MB. The RAID-1 volume is named d30.

filesys
The /usr file system is created and mirrored on the slices c0t1d0s4 and c0t0d0s4. The size of the /usr file system is set to 4096 MB. The RAID-1 volume is named d40.

metadb
Three state database replicas (metadbs) are installed on slice c0t1d0s7, and are sized at 8192 blocks (4 MB).

■ For overview information about how to create mirrored file systems during your installation, see Chapter 8, “Creating RAID-1 Volumes (Mirrors) During Installation (Overview),” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.

■ For guidelines and requirements of creating mirrored file systems, see Chapter 9, “Creating RAID-1 Volumes (Mirrors) During Installation (Planning),” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.

■ For descriptions of keyword values, see “filesys Profile Keyword (Creating RAID-1 Volumes)” on page 122 and “metadb Profile Keyword (Creating State Database Replicas)” on page 129.

EXAMPLE 3–13 Creating a RAID-1 Volume to Mirror the Root File System

In the following example, the profile indicates that the JumpStart program uses Solaris Volume Manager technology to create a RAID-1 volume (mirror) for the root (/) file system.

```plaintext
# profile keywords profile values
# ---------------- -------------------
install_type: initial_install
cluster: SUNWcXall
filesys: mirror:d30 c0t1d0s0 c0t0d0s0 /
filesys: c0t0d0s3 512 swap
metadb: c0t0d0s4 size 8192 count 4
metadb: c0t1d0s4 size 8192 count 4
```

Some of the keywords and values in this example are as follows:
EXAMPLE 3–13 Creating a RAID-1 Volume to Mirror the Root File System  (Continued)

cluster The Entire Oracle Solaris Software Plus OEM Support software group, SUNWCXall, is installed on the system.

filesys The root (/) file system is created and mirrored on the slices c0t1d0s0 and c0t0d0s0. The RAID-1 volume that mirrors c0t1d0s0 and c0t0d0s0 is named d100. The JumpStart program assigns names to the two submirrors.

filesys The swap file system is created and mirrored on the slice c0t0d0s3, and is sized at 512 MB.

metadb Four state database replicas (metadbs) are installed on slice c0t0d0s4, and are sized at 8192 blocks (4 MB).

metadb Four state database replicas (metadbs) are installed on slice c0t1d0s4, and are sized at 8192 blocks (4 MB).

EXAMPLE 3–14 Creating RAID-1 Volumes to Mirror Multiple File Systems

In the following example, the profile indicates that the JumpStart program uses Solaris Volume Manager technology to create RAID-1 volumes (mirrors) for the root (/), swap, and /usr file systems.

```
# profile keywords     profile values
# ---------------- -------------------
install_type           initial_install
cluster                SUNWCXall
filesys                mirror:d100 c0t1d0s0 c0t0d0s0 200 /
filesys                c0t1d0s5 500 /var
filesys                c0t0d0s5 500
filesys                mirror c0t0d0s1 512 swap
metadb                 c0t0d0s3 size 8192 count 5
filesys                mirror c0t1d0s4 c0t0d0s4 2000 /usr
filesys                c0t1d0s7 free /export/home
filesys                c0t0d0s7 free
```

Some of the keywords and values from in this example are as follows:

cluster The Entire Oracle Solaris Software Plus OEM Support software group, SUNWCXall, is installed on the system.

filesys The root (/) file system is created and mirrored on the slices c0t1d0s0 and c0t0d0s0. The size of the root (/) file system is set to 200 MB. The RAID-1 volume that mirrors c0t1d0s0 and c0t0d0s0 is named d100.

filesys The /var file system is installed on the slice c0t1d0s5 and is sized at 500 MB. The root (/) file system is created and mirrored on the slices c0t1d0s0 and c0t0d0s0. The size of the root (/) file system is set to 200 MB. The RAID-1 volume that mirrors c0t1d0s0 and c0t0d0s0 is named d100.
filesys The swap file system is created and mirrored on the slice c0t0d0s1, and is sized at 512 MB. The JumpStart program assigns a name to the mirror.

metadb Five state database replicas (metadbs) are installed on slice c0t0d0s3, and are sized at 8192 blocks (4 MB).

filesys The /usr file system is created and mirrored on the slices c0t1d0s4 and c0t0d0s4. The size of the /usr file system is set to 2000 MB. The JumpStart program assigns a name to the mirror.

EXAMPLE 3–15 x86: Using the fdisk Keyword

<table>
<thead>
<tr>
<th># profile keywords</th>
<th>profile values</th>
</tr>
</thead>
<tbody>
<tr>
<td># profile keywords</td>
<td>profile values</td>
</tr>
<tr>
<td>system_type</td>
<td>standalone</td>
</tr>
<tr>
<td>fdisk</td>
<td>c0t0d0 0x04 delete</td>
</tr>
<tr>
<td>fdisk</td>
<td>c0t0d0 solaris maxfree</td>
</tr>
<tr>
<td>cluster</td>
<td>SUNWCall</td>
</tr>
<tr>
<td>cluster</td>
<td>SUNWCacc delete</td>
</tr>
</tbody>
</table>

Some of the keywords and values in this example are as follows:

fdisk All fdisk partitions of type DOSOS16 (04 hexadecimal) are deleted from the c0t0d0 disk.

fdisk An Oracle Solaris fdisk partition is created on the largest contiguous free space on the c0t0d0 disk.

cluster The Entire Distribution software group, SUNWCall, is installed on the system.

cluster The system accounting utilities, SUNWCacc, are not to be installed on the system.

EXAMPLE 3–16 Creating a JumpStart Profile for An Initial iSCSI Installation

install_type initial_install
partitioning explicit
filesys rootdisk.s4 5000
filesys rootdisk.s1 2048
iscsi_target_name iqn.1986-03.com.sun:02:358ddbf8-601a-e73a-df56-89
iscsi_target_ip 10.12.162.24
iscsi_target_lun 0
iscsi_target_port 3260
cluster SUNWCrnet

The following example shows a sample JumpStart profile for an upgrade installation.
EXAMPLE 3–17 JumpStart Profile For An iSCSI Upgrade Installation

install_type upgrade
iscsi_target_name iqn.1986-03.com.sun:02:358dbf8-601a-e73a-df56-89
iscsi_target_ip 10.12.162.24
iscsi_target_lun 0
iscsi_target_port 3260
iscsi_target_slice 4

See “iSCSI Profile Keyword” on page 131 for details about the keywords used in the JumpStart profile for iSCSI upgrade and installation.

Caution – The following keywords are used in the JumpStart profile to indicate where the locations of the directories such as the root directory, swap space, the /usr directory and so on have to be created.

- filesys
- pool
- boot_device

Since the disk number is dynamically generated during an iSCSI installation, the user should not explicitly provide the disk nomenclature (cXtXdXsX) for the above keywords in the JumpStart profile during an iSCSI installation.

Caution – When the iSCSI key-value information is provided in the profile, the boot_device key-value pair and root_device key-value pair should not be included in the profile. The root_device key-value pair is dynamically assigned by the pfinstall command based on the iSCSI target.

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 whether 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 Oracle 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 59 and "x86: How to Create a Multiple-Disk Configuration File" on page 64.

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

How to Create a Temporary Oracle Solaris Environment to Test a Profile

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

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

- You want to test an Oracle Solaris 10 8/11 upgrade profile on a system that is running a previous version of the Oracle Solaris software.
- You do not have an Oracle Solaris 10 8/11 system installed yet to test Oracle Solaris 10 8/11 initial installation profiles.

1. Boot a system from an image of one of the following:
   For SPARC based systems:
   - Oracle Solaris Operating System for SPARC Platforms DVD
   - Oracle Solaris Software for SPARC Platforms - 1 CD
   For x86 based systems:
   - Oracle Solaris Operating System for x86 Platforms DVD
   - Oracle Solaris Software for x86 Platforms - 1 CD

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

2. Respond to the system identification questions.
Exit from the installation program by typing ! at the prompt.
The Solaris installation program will assist you in installing software for Solaris.<Press ENTER to continue> ("!" exits)

Execute the pfinstall command from the shell.
For details about using the pfinstall command, see Step 5 in “How to Test a Profile” on page 46.

How to Test a Profile

x86 only – If you are using the locale keyword, the pfinstall -D command fails to test the profile. For a workaround, see the error message "could not select locale," in the section, “Upgrading the Oracle Solaris OS” in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations.

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 Test the profile.

- To test an initial installation profile if you have a system that is running the Oracle Solaris 10 8/11 software, become superuser on the system.

- To test an upgrade profile, or if you do not have a system that is running Oracle Solaris 10 8/11 to test an initial installation profile:

  a. Create a temporary Oracle Solaris 10 8/11 environment to test the profile. For details, see “How to Create a Temporary Oracle Solaris Environment to Test a Profile” on page 45. Then, go to Step 3.

  b. Create a temporary mount point.

    # mkdir /tmp/mnt

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

    - If you are mounting a remote NFS file system for systems on the network type the following:

    ```
    mount -F nfs server-name:path /tmp/mnt
    ```
To test the profile with a specific system memory size, set SYS_MEMSIZE to the specific memory size in MB.

```
# SYS_MEMSIZE=memory-size
# export SYS_MEMSIZE
```

Mount the directory.

- If you mounted a directory in Substep c, change the directory to /tmp/mnt.
  ```
  # cd /tmp/mnt
  ```
- If you did not mount a directory, change to the directory to where the profile is located, which is usually the JumpStart directory.
  ```
  # cd jumpstart-dir-path
  ```

5 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 Oracle Solaris 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 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 about how to create a disk configuration file, see “Creating Disk Configuration Files” on page 59.

**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 and use the `-D` option.
The path to the Oracle Solaris software image. You use this option, for example, if the system is using Solaris Volume Manager to mount the Oracle Solaris Software - 1 CD for your platform.

**Profile Test Example**

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 Oracle Solaris 10 8/11 software is installed. The `basic_prof` profile is located in the `/jumpstart` directory, and the path to the Oracle Solaris Operating System DVD image is specified because Solaris Volume Manager is being used.

**EXAMPLE 3–18** Profile Test Using an Oracle Solaris 10 8/11 System

```bash
# 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 an Oracle Solaris 10 8/11 system. The test is performed against the 535_test disk configuration file. The test checks for 64 MB of system memory. This example uses an Oracle Solaris Software for SPARC Platforms - 1 CD or Oracle Solaris Software for x86 Platforms - 1 CD image that is located in the `/export/install` directory.

**EXAMPLE 3–19** Profile Test Using a Disk Configuration File

```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 JumpStart installation software to match a system to a profile.

The following table 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 begin, class, and finish fields are specified for each rule. The begin and finish fields can consist of a minus sign ( - ) 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, check creates the rules.ok file from the rules file, removes all comments and blank lines, retains all rules, and adds the following comment line at the end: # version=2 checksum=num</td>
</tr>
</tbody>
</table>

▶ How 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_10/Misc/jumpstart_sample directory on the Oracle Solaris Operating System DVD or on the Oracle Solaris Software - 1 CD.

2 Change the directory to the JumpStart directory.

3 Run the check script to validate the rules file:

   $ ./check -p path -r file-name

   -p path Validates the rules by using the check script from the Oracle Solaris software image instead of the check script from the system you are using. path is the image on a local disk or a mounted Oracle Solaris Operating System DVD or an Oracle Solaris Software - 1 CD.

   Use this option to run the most recent version of check if your system is running a previous version of Oracle Solaris.
Validating the rules File

- \textit{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 following information.

\begin{itemize}
  \item The JumpStart configuration is ok
\end{itemize}

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

See Also After you validate the rules file, you can learn more about optional JumpStart features in Chapter 4, “Using Optional JumpStart Features (Tasks).” You can learn about performing JumpStart installations in Chapter 6, “Performing a JumpStart Installation (Tasks).”
This chapter describes the optional features that are available to create additional JumpStart installation tools.

**Note** – If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, "Installing a ZFS Root Pool With JumpStart," for limitations and profile examples.

This chapter covers the following topics:

- "Creating Begin Scripts" on page 51
- "Creating Finish Scripts" on page 54
- "Creating a Compressed Configuration File" on page 58
- "Creating Disk Configuration Files" on page 59
- "Using a Site-Specific Installation Program" on page 65

**Note** – Instructions in this chapter are valid for either a SPARC server or an x86 server that is being used to provide JumpStart files, called a *profile server*. A profile server can provide JumpStart files for different platform types. For example, a SPARC server can provide JumpStart files for both SPARC based systems and x86 based 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 Oracle Solaris software is installed on a system. You can use begin scripts only when using JumpStart to install the Oracle Solaris software.

Use a begin script to perform the following tasks:

- Creating derived profiles
- Backing up files before upgrading
- Record the duration of an installation

### 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 JumpStart environment variables in your begin scripts. For a list of 
  environment variables, see "JumpStart Environment Variables" on page 145.

- Save begin scripts in the JumpStart directory.

---

**Note** – For the Oracle Solaris 10 release, a sample JumpStart script, `set_nfs4_domain`, was 
provided on media to prevent being prompted during a JumpStart installation. This script 
suppressed the NFSv4 prompt during installation. This script is no longer required. Starting 
with the Solaris 10 5/09 release, use the `sysidcfg` keyword `nfs4_domain` to suppress the 
prompt. The `set_nfs4_domain` script no longer works to suppress a prompt.

If you have non-global zones installed and the new `nfs4_domain` keyword exists in the 
`sysidcfg` file, the first boot of a non-global zone sets the domain. Otherwise, the Oracle Solaris 
interactive installation program starts and you are prompted to provide a domain name before 
the boot process completes.

See "nfs4_domain Keyword" in Oracle Solaris 10 1/13 Installation Guide: Network-Based 
Installations

---

### Creating Derived Profiles With a Begin Script

A derived profile is a profile that is dynamically created by a begin script during a 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 Oracle 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 Oracle Solaris software on the system.

The following example shows 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 4–1  Begin Script That Creates a Derived Profile**

```
#!/bin/sh

echo

"install_type initial_install" > ${SI_PROFILE}

"system_type standalone" >> ${SI_PROFILE}

"partitioning default" >> ${SI_PROFILE}

"cluster SUNWCprog" >> ${SI_PROFILE}

"package SUNWman delete" >> ${SI_PROFILE}

"package SUNWolman delete" >> ${SI_PROFILE}

"package SUNWXwman delete" >> ${SI_PROFILE}
```

In the example, the begin script must use the SI_PROFILE environment variable for the name of the derived profile, which is set to /tmp/install.input by default.

---

**Note** - If a begin script is used to create a derived profile, ensure the script does not have any errors. A derived profile is not verified by the check script because derived profiles are not created until the execution of the begin script.

---

**Tracking Installation Duration With a Begin Script and Finish Script**

You can include a begin script and a finish script to track the start time and end time of an installation, as shown in the following examples.

**EXAMPLE 4–2  Begin Script That Tracks Starting Time**

```
#!/bin/sh

# more begin-with-date

"Noting time that installation began in /tmp/install-begin-time"

"Install begin time: 'date'" > /tmp/install-begin-time

"cat /tmp/install-begin-time"

#
```

**EXAMPLE 4–3  Finish Script That Tracks Ending Time**

```
#!/bin/sh

# more finish*with*date
```

---

Creating Begin Scripts
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 Oracle Solaris software is installed on a system but before the system reboots. You can use finish scripts only when using JumpStart to install Oracle Solaris.

Tasks that you can perform with a finish script include the following:

- Adding files
- Adding individual packages or patches in addition to the ones that are installed in a particular software group
- Customizing the root environment
- Installing additional software

About Finish Scripts

- The Oracle Solaris installation 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 JumpStart environment variables in your finish scripts. For a list of environment variables, see “JumpStart Environment Variables” on page 145.
- Save finish scripts in the JumpStart directory.

EXAMPLE 4–3  Finish Script That Tracks Ending Time  (Continued)

# cp /tmp/install-begin-time /a/var/tmp
echo "Noting time that installation finished in /a/var/tmp/install-finish-time"
echo "Install finish time: 'date'" > /a/var/tmp/install-finish-time
echo
cat /a/var/tmp/install-finish-time
#

The start and end times will be recorded in the finish.log file.
In the past, the `chroot(1M)` command was used with the `pkgadd` and `patchadd` commands in the finish script environment. In rare instances, some packages or patches do not work with the `-R` option. You must create a dummy `/etc/mnttab` file in the `/a` root path before issuing the `chroot` command.

To create a dummy `/etc/mnttab` file, add the following line to your finish script:

```
cp /etc/mnttab /a/etc/mnttab
```

### Adding 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.

After you copy all of the files that you are adding to the installed system to the JumpStart directory, insert the following line in the finish script for each file that you want to be copied to the newly installed file system hierarchy:

```
cp ${SI_CONFIG_DIR}/filename /a/path
```

**EXAMPLE 4-4** Adding a File With a Finish Script

Assume you have a special application, `site_prog`, developed for all users at your site. If you place a copy of `site_prog` into the JumpStart directory, the following line in a finish script copies `site_prog` from the JumpStart directory into a system’s `/usr/bin` directory:

```
cp ${SI_CONFIG_DIR}/site_prog /a/usr/bin
```

### Adding Packages or Patches With a Finish Script

You can create a finish script to automatically add packages or patches after the Oracle Solaris software is installed on a system. By adding packages with a finish script, you reduce time and ensure consistency in which 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 4–5** shows an example of a finish script that adds packages.
- **Example 4–6** shows an example of a finish script that adds patches.
EXAMPLE 4–5 Adding Packages With a Finish Script

#!/bin/sh

BASE=/a
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin

mkdir ${MNT}
mount -f nfs sherlock:/export/package ${MNT}
cat >${ADMIN_FILE} <<DONT_ASK
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} SUNWxyz
umount ${MNT}
rmdir ${MNT}

The commands for this example are as follows:

- The following command mounts a directory on a server that contains the package to install.
  ```bash
  mount -f nfs sherlock:/export/package ${MNT}
  ```
- The following command creates a temporary package administration file, admin, to force the pkgadd(1M) command not to perform checks or prompt for questions when installing a package. Use the temporary package administration file to maintain a hands-off installation when you are adding packages.
  ```bash
  cat >${ADMIN_FILE} <<DONT_ASK
  mail=root
  instance=overwrite
  partial=nocheck
  runlevel=nocheck
  idepend=nocheck
  rdepend=nocheck
  space=ask
  setuid=nocheck
  conflict=nocheck
  action=nocheck
  basedir=default
  DONT_ASK
  ```
- The following pkgadd command adds the package by using the -a option, specifying the package administration file, and the -R option, specifying the root path.
  ```bash
  /usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNWxyz
  ```

EXAMPLE 4–6 Adding Patches With a Finish Script

#!/bin/sh

########
# USER-CONFIGURABLE OPTIONS
#
########

# The location of the patches to add to the system after it’s installed.
# The OS rev (5.x) and the architecture ('mach') will be added to the
EXAMPLE 4-6  Adding Patches With a Finish Script  

(Continued)

```bash
# root. For example, /foo on a 8 SPARC would turn into /foo/5.8/sparc
LUPATCHHOST=ins3525-svr
LUPATCHPATHROOT=/export/solaris/patchdb
#########
# NO USER-SERVICEABLE PARTS PAST THIS POINT
#
#########
BASEDIR=/a

# Figure out the source and target OS versions
echo Determining OS revisions...
SRCREV='uname -r'
echo Source $SRCREV
LUPATCHPATH=$LUPATCHPATHROOT/$SRCREV/
mach

# Add the patches needed
# echo Adding OS patches
mount $LUPATCHHOST:$LUPATCHPATH /mnt >/dev/null 2>&1
if [ $? = 0 ]; then
  for patch in
    cat /mnt/*Recommended/patch_order
  do
    (cd /mnt/*Recommended/$patch ; echo yes | patchadd -u -d -R $BASEDIR .)
  done
  cd /tmp
  umount /mnt
else
  echo "No patches found"
if
```

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 the following example customizes the root environment by appending information to the .cshrc file in the root (/) directory.

EXAMPLE 4-7  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@\$host -n> "
  alias cp cp -i
```
Non-Interactive Installations With Finish Scripts

You can use finish scripts to install additional software after the Oracle Solaris OS is installed. The Oracle Solaris installation program prompts you to enter information during the installation. To maintain a hands-off installation, you can run the Oracle Solaris installation program with the `-nodisplay` or `-noconsole` options.

TABLE 4–1 Oracle Solaris Installation Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-nodisplay</code></td>
<td>Runs the installer without a graphic user interface. Uses the default product installation unless the installation was modified by the <code>-locales</code> option.</td>
</tr>
<tr>
<td><code>-noconsole</code></td>
<td>Runs the installation without any interactive text console device. Useful when paired with <code>-nodisplay</code> for UNIX script use.</td>
</tr>
</tbody>
</table>

For more information, see the `installer(1M)` man page.

Creating a Compressed Configuration File

Instead of using the `add_install_client` command to specify the location of the 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. As a result, you must compress all of the JumpStart configuration files into one file.

- **For SPARC based systems**, you specify the location of the file in the `boot` command
- **For x86 based systems**, you specify the location of the files by editing the GRUB entry in the GRUB menu

The compressed configuration file can be one of the following types:

- `tar`
- `Compressed tar`
How 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 JumpStart configuration files into one file.
   
   The compressed configuration file cannot contain relative paths. The 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 a local hard disk.
   
   The following example shows how to use the `tar` command to create a compressed configuration file that is named `config.tar`. The JumpStart configuration files are located in the `/jumpstart` directory.

   ```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: How to Create a Single-Disk Configuration File

1. Locate a SPARC based system with a disk you want to test.

2. Become superuser or assume an equivalent role.
   Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in System Administration Guide: Security Services.

3. Redirect the output of the `prtvtoc(1M)` command to a file.
   ```
   # prtvtoc /dev/rdisk/device-name > disk-config-file
   /dev/rdisk/device-name  The device name of the system’s disk. `device-name` must be in the form `cwx dys` or `cx dys`.
   disk-config-file  The name of the disk configuration file.
   ```

Example 4–9  SPARC: Creating a Disk Configuration File

The following example shows how to create a single-disk configuration file, `104_test`, on a SPARC based system with a 104 MB disk.

Redirect the output of the `prtvtoc` command to a single-disk configuration file that is named `104_test`:

```
# prtvtoc /dev/rdisk/c0t3d0s2 >104_test
```

The contents of the `104_test` file resemble the following example:

```
* /dev/rdisk/c0t3d0s2 partition map
*
* Dimensions:
*  512 bytes/sector
*  72 sectors/track
*  14 tracks/cylinder
*  1008 sectors/cylinder
*  280 cylinders*  2036 accessible cylinders
* Flags:
*  1: unmountable
*  10: read-only
*
* Partition  Tag  Flags  First Sector  Count  Last Sector  Mount Directory
  1   2   00  0  164304  164303  /
  2   5   00  0  2052288  2052287
  3   0   00  164304  823536  987839  /disk2/b298
  5   0   00  987840  614880  1602719  /install/298/sparc/work
  7   0   00  1602720  449568  2052287  /space
```
SPARC: How to Create a Multiple-Disk Configuration File

1. Locate a SPARC based system with a disk you want to test.

2. Become superuser or assume an equivalent role.
   Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in System Administration Guide: Security Services.

3. Redirect the output of the `prtvtoc(1M)` command to a file.
   ```
   # prtvtoc /dev/rdsk/device-name >disk-config-file
   /dev/rdsk/device-name The device name of the system’s disk. `device-name` must be in the form `cwtxdys2` or `cx dys2`.
   disk-config-file The name of the disk configuration file.
   ```

4. Concatenate the single-disk configuration files and save the output in a new file.
   ```
   # cat disk-file1 disk-file2 >multi-disk-config-file
   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. If the target numbers in the disk device names are not unique in the multiple-disk configuration file, 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
   ```
x86: How to Create a Single-Disk Configuration File

1. Locate an x86 based system that contains a disk that you are testing.

2. Become superuser or assume an equivalent role.
   Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in System Administration Guide: Security Services.

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 `cwxtydxys0` or `cx dys0`.
   ```

4. Append the output of the `prtvtoc(1M)` command to the disk configuration file:
   ```bash
   # prtvtoc /dev/rdsk/device-name >> disk-config
   
   /dev/rdsk/device-name The device name of the system's disk. device-name must be in the form `cwxtydxys2` or `cx dys2`.
   
   disk-config The name of the disk configuration file.
   ```

Example 4–10  x86: Creating a Disk Configuration File

The following example shows how to create a single-disk configuration file, `500_test`, on an x86 based system that contains a 500 MB disk.

First, save the output of the `fdisk` command to a file that is named `500_test`:

```bash
# fdisk -R -W 500_test -h /dev/rdsk/c0t0d0p0
```

The `500_test` file looks like the following:

```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
```
Append the output of the prtvtoc command to the 500_test file:

```
# prtvtoc /dev/rdsk/c0t0d0s2 >>500_test
```

The 500_test file is now a complete disk configuration file:

```
* /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: EXTDOS
* 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
```

Creating Disk Configuration Files

Chapter 4 • Using Optional JumpStart Features (Tasks) 63
x86: How to Create a Multiple-Disk Configuration File

1. Locate an x86 based system that contains a disk that you are testing.

2. Become superuser or assume an equivalent role.
   Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in System Administration Guide: Security Services.

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 `cwt.x dys0` or `cx dys0`.
   
   # prtvtoc /dev/rdsk/device-name >> disk-config
   
   /dev/rdsk/device-name The device name of the system’s disk. `device-name` must be in the form `cwt.x dys2` or `cx dys2`.
   
   disk-config The name of the disk configuration file.
   
   # cat disk-file1 disk-file2 > multi-disk-config-file
   
   The new file becomes the multiple-disk configuration file, as in the following example.
   
   # cat 104_disk2 104_disk3 104_disk5 > multi_disk_test
   ```
If the target numbers in the disk device names are not unique in the multiple-disk configuration file, make the target numbers unique.

For example, the file might contain 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

Using a Site-Specific Installation Program

You can also use begin and finish scripts to create your own installation program to install Oracle Solaris software.

When you specify a minus sign (-) in the profile field, begin and finish scripts control how Oracle Solaris software is installed on a system instead of the profile and the Oracle Solaris installation program.

For example, if the following rule matches a system, the x_install.beg begin script and the x_install.fin finish script install Oracle 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.

**Note** – If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, “Installing a ZFS Root Pool With JumpStart,” for limitations and profile examples.

This chapter covers the following topics:

- “Probe Keywords” on page 67
- “Creating a custom_probes File” on page 68
- “Validating the custom_probes File” on page 70

**Probe Keywords**

A probe keywords are related to rule keywords. A rule keyword is a predefined lexical unit or word that describes a general system attribute, such as hostname, 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 Oracle Solaris software is to be installed on each system in the group.

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 ST_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 147.
Creating a custom_probes File

The rule and probe keywords that are described in “Rule Keywords and Values” on page 97 and “Probe Keywords and Values” on page 147 might not be 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_env 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.

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 arguments are interpreted in sequence. The sequence starts after the keyword and ends 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 the custom_probes File

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_tcb defines the new probe keyword tcb. Functions that begin with cmp_ define new rule keywords. For example, cmp_tcb defines the new rule keyword tcb.
How to Create a custom_probes File

1. Create a Bourne shell script text file and name it `custom_probes`.

2. In the `custom_probes` text file, define your probe and comparison functions.

   When you use 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. The sequence starts after the keyword and ends 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.

Example 5–1  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 [ "$SI_TCX" = "$1" ]; then
    return 0
  else
    return 1
  fi
}
```

Example 5–2  Custom Probe Keyword Used in a rules File

This 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.
Validating the custom probes File

```plaintext
probe tcx
tcx tcx - profile_tcx -
any any - profile -
```

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 Oracle Solaris software installed
- `/Solaris_10/Tools/Boot/usr/sbin/install.d/chkprobe` on the Oracle Solaris Operating System DVD or on the Oracle Solaris Software - 1 CD

**Note** – Always place probe keywords at or near the beginning of the rules file. This placement ensures that the keywords are read and run before other rule keywords that might rely on the probe keywords.

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.

When you use the check script, the following process occurs:

1. check searches for a custom_probes file.
2. 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:

   ```plaintext
   # version=2 checksum=num
   ```

How 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_10/Misc/jumpstart_sample directory on the Oracle Solaris Operating System DVD or on the Oracle Solaris Software - 1 CD.

2. **Change to the JumpStart directory.**

3. **Run the check script to validate the rules and custom_probes files.**

   ```bash
   $ ./check -p path -r file-name
   ```
Validating the `custom_probes` File

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-p path</code></td>
<td>Validates the <code>custom_probes</code> file by using the check script from the Oracle Solaris software image for your platform instead of the check script from the system you are using. <em>path</em> is the image on a local disk or a mounted Oracle Solaris Operating System DVD or Oracle Solaris Software - 1 CD. Use this option to run the most recent version of check if your system is running a previous version of Oracle Solaris.</td>
</tr>
<tr>
<td><code>-r file-name</code></td>
<td>Specifies a file name other than the one that is named <code>custom_probes</code>. By using the <code>-r</code> option, you can test the validity of a set of functions before integrating the functions into the <code>custom_probes</code> file.</td>
</tr>
</tbody>
</table>

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 JumpStart configuration is ok* and creates the `rules.ok` and `custom_probes.ok` files in the JumpStart directory.

4. If the `custom_probes.ok` file is not executable, type the following command:
   ```
   # chmod +x custom_probes
   ```

5. Ensure that `root` owns the `custom_probes.ok` file and that the permissions are set to 755.
Performing a JumpStart Installation (Tasks)

This chapter describes how to perform a 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 Oracle Solaris software.

Note – If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, "Installing a ZFS Root Pool With JumpStart," for limitations and profile examples.

Note – If you are installing or upgrading the Oracle Solaris 10 OS on an iSCSI target system, see the following sections for iSCSI keywords and examples.

- "iSCSI Profile Keyword" on page 131
- Example 3–16
- Example 3–17

For detailed instructions about configuring iSCSI parameters, see Chapter 4, "Installing the Oracle Solaris 10 OS on an iSCSI Target Disk," in Oracle Solaris 10 1/13 Installation Guide: Basic Installations.

This chapter covers the following topics:

- "JumpStart Installation Issues" on page 74
- "SPARC: Setting Up a System for a JumpStart Installation (Task Map)" on page 75
- "SPARC: Performing a JumpStart Installation" on page 77
- “x86: Setting Up a System for a JumpStart Installation (Task Map) ” on page 80
- “x86: Performing a JumpStart Installation” on page 82
JumpStart Installation Issues

You should be aware of some issues before performing a JumpStart installation. Review the following table for specific information.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>The sampleJumpStartscript is no longer required to suppress the NFSv4 prompt</td>
<td>For the Solaris 10 release, a sample JumpStart script, <code>set_nfs4_domain</code>, was provided on media to prevent being prompted during a JumpStart installation. This script suppressed the NFSv4 prompt during installation. This script is no longer required. <strong>Starting with the Solaris 10 8/07 release,</strong> use the <code>sysidcfg</code> keyword, <code>nfs4_domain</code> to suppress the prompt. The <code>set_nfs4_domain</code> script no longer works to suppress a prompt. If you have non-global zones installed and the new <code>nfs4_domain</code> keyword exists in the <code>sysidcfg</code> file, the first boot of a non-global zone sets the domain. Otherwise, the Oracle Solaris interactive installation program starts and you are prompted to provide a domain name before the boot process completes.</td>
<td>&quot;nfs4_domain Keyword&quot; in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations</td>
</tr>
<tr>
<td>Providing Auto Registrationinformation in the <code>sysidcfg</code> file keeps JumpStart installations hands-off</td>
<td>Auto Registration, a feature of Oracle Solaris, is new in the Oracle Solaris 10 9/10 release. When you install or upgrade your system, configuration data about your system is, on rebooting, automatically communicated through the existing service tag technology to the Oracle Product Registration System. This service tag data about your system is used, for example, to help Oracle enhance customer support and services. If you include the <code>auto_reg</code> keyword in the <code>sysidcfg</code> file prior to the installation or upgrade, the installation can remain completely hands-off. However, if you do not include the <code>auto_reg</code> keyword, you will be prompted to provide your support credentials and proxy information for Auto Registration during the installation or upgrade.</td>
<td></td>
</tr>
<tr>
<td>Selecting a keyboard language in the <code>sysidcfg</code> file prevents a prompt</td>
<td>If your keyboard is not self-identifying and you want to prevent being prompted during your JumpStart installation, select the keyboard language in your <code>sysidcfg</code> file. For JumpStart installations, the default is for the U.S. English language. To select another language and its corresponding keyboard layout, set the keyboard keyword in your <code>sysidcfg</code> file.</td>
<td>■ &quot;sysidcfg File Keywords&quot; in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations ■ For man pages, see: ■ <code>sysidtool(1M)</code> ■ <code>sysidcfg(4)</code></td>
</tr>
</tbody>
</table>
TABLE 6–1  JumpStart Installation Limitations  (Continued)

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you have non-global zones, use Live Upgrade to upgrade</td>
<td>You can upgrade a system that has non-global zones installed with JumpStart, but Live Upgrade is the recommended program to upgrade. JumpStart might require extensive upgrade time, because the time required to complete the upgrade increases linearly with the number of installed non-global zones.</td>
<td>Oracle Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning</td>
</tr>
<tr>
<td>A flash archive cannot contain non-global zones</td>
<td>If you use a flash archive to install an archive that contains non-global zones is not properly installed on your system.</td>
<td>For general information about creating non-global zones, see System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones.</td>
</tr>
<tr>
<td>A flash archive cannot be properly created when a non-global zone is installed.</td>
<td>The Flash Archive feature is not compatible with the Oracle Solaris Zones partitioning technology. If you create a flash archive, the resulting archive is not installed properly when the archive is deployed under these conditions:  ■ The archive is created in a non-global zone  ■ The archive is created in a global zone that has non-global zones installed</td>
<td></td>
</tr>
<tr>
<td>When using the archive_location keyword to install a flash archive, the archive and the installation media must contain identical operating system versions.</td>
<td>For example, if the archive is an Oracle Solaris 10 8/11 operating system and you are using DVD media, then you must use Oracle Solaris 10 8/11 DVD media to install the archive. If the operating systems versions do not match, the installation on the clone system fails.</td>
<td></td>
</tr>
<tr>
<td>SPARC: Additional hardware requirements</td>
<td>Refer to your hardware documentation for any additional requirements for your platform that might be required to complete a JumpStart installation.</td>
<td></td>
</tr>
</tbody>
</table>

---

**SPARC: Setting Up a System for a JumpStart Installation (Task Map)**

TABLE 6–2  Task Map: Setting Up a System for a JumpStart Installation

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check whether the system is supported.</td>
<td>Check the hardware documentation for system support in the Oracle Solaris environment.</td>
<td>Oracle Solaris Sun Hardware Platform Guide at <a href="http://www.oracle.com/technetwork/indexes/documentation/index.html">http://www.oracle.com/technetwork/indexes/documentation/index.html</a></td>
</tr>
</tbody>
</table>
### TABLE 6-2  Task Map: Setting Up a System for a JumpStart Installation  (Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check whether the system has enough disk space for the Oracle Solaris software.</td>
<td>Verify that you have planned enough space to install the Oracle Solaris software on your system.</td>
<td>Chapter 3, “System Requirements, Guidelines, and Upgrade Information,” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</td>
</tr>
<tr>
<td>(Optional) Set system parameters.</td>
<td>You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.</td>
<td>Chapter 2, “Preconfiguring System Configuration Information (Tasks),” in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations</td>
</tr>
<tr>
<td>Prepare the system for JumpStart installation.</td>
<td>Create and validate a rules file and profile files.</td>
<td>Chapter 3, “Preparing JumpStart Installations (Tasks)”</td>
</tr>
<tr>
<td>(Optional) Prepare optional JumpStart features.</td>
<td>If you are using begin scripts, finish scripts, or other optional features, prepare the scripts or files.</td>
<td>Chapter 4, “Using Optional JumpStart Features (Tasks);” and Chapter 5, “Creating Custom Rule and Probe Keywords (Tasks)”</td>
</tr>
<tr>
<td>(Optional) Prepare to install the Oracle Solaris software from the network.</td>
<td>To install a system from a remote Oracle Solaris Operating System DVD or Oracle Solaris Software for SPARC Platforms CD image, you need to set up the system to boot and install from an install server or a boot server.</td>
<td>Chapter 5, “Installing From the Network With DVD Media (Tasks),” in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations Chapter 6, “Installing From the Network With CD Media (Tasks),” in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations</td>
</tr>
<tr>
<td>(Optional) Prepare for a Flash Archive installation.</td>
<td>Set up specifics for a Flash Archive installation.</td>
<td>“To Prepare to Install a Flash Archive With a JumpStart Installation” on page 77</td>
</tr>
<tr>
<td>Perform an installation or upgrade.</td>
<td>Boot the system to initiate the installation or upgrade.</td>
<td>“SPARC: To Perform an Installation or Upgrade With the JumpStart Program” on page 78</td>
</tr>
</tbody>
</table>
SPARC: Performing a JumpStart Installation

During a 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. When 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.

You can install a full archive for an initial installation or if you have already installed an archive, a differential archive for an update. You can use the JumpStart installation method or use Live Upgrade to install an archive on an inactive boot environment. For an overview of a full or differential archive, see Chapter 1, “Flash Archive Overview,” in Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation).

▼ To Prepare to Install a Flash Archive With a JumpStart Installation

This procedure provides the instructions to install an archive with JumpStart. For procedures about installing an archive on an inactive boot environment by using Live Upgrade, see “How to Install a Flash Archive With a Profile” in Oracle Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning.

Before You Begin

Review the limitations on flash archive installations in Table 6–1

1. **On the install server, create the JumpStart rules file.**
   For detailed instructions about creating JumpStart files, refer to Chapter 3, "Preparing JumpStart Installations (Tasks)."

2. **On the install server, create the JumpStart profile file.**
   For examples of Flash Archive profiles, see “Profile Examples” on page 33.
   For information about the only keywords valid when you install a Flash Archive, see Table 8–2,
   a. **Set the value of the keyword **install_type**.**
      - 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 flash archive by using the **archive_location** keyword.**
      For details about the `archive_location` keyword, refer to “archive_location Keyword” on page 103.
c. Specify the file system configuration.
   The Flash Archive extraction process does not support auto-layout of partitions.

d. (Optional) If you want to install additional packages at the same time you install an archive, use the package keyword.
   For more information, see “package Profile Keyword (UFS and ZFS)” on page 132.

e. (Optional) If you want to install an additional 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 flash archive.
   For detailed instructions, refer to the following:
   ■ “Adding Systems to Be Installed From the Network With a DVD Image” in Oracle Solaris 10/13 Installation Guide: Network-Based Installations
   ■ “Adding Systems to Be Installed From the Network With a CD Image” in Oracle Solaris 10/13 Installation Guide: Network-Based Installations

4 Perform the JumpStart installation on the clone systems.
   For detailed instructions, refer to “SPARC: To Perform an Installation or Upgrade With the JumpStart Program” on page 78.

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

Before You Begin
   ■ If the system is part of a network, ensure that an Ethernet connector or similar network adapter is attached to your system.
   ■ 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.

1 If you are using the system’s DVD-ROM or CD-ROM drive to install the Oracle Solaris software, insert the Oracle Solaris Operating System for SPARC Platforms DVD or the Oracle Solaris Software for SPARC Platforms - 1 CD in the drive.

2 If you are using a profile diskette, insert the profile diskette in the system’s diskette drive.

3 Boot the system.
   ■ If the system is new and unmodified, turn on the system.
If you want to install or upgrade an existing system, shut down the system. At the ok prompt, type the appropriate options for the boot command. The syntax of the boot command is the following.

`ok boot [cd-dvd|net] - install [url|ask] options`

For example, if you type the following command, the OS is installed over the network by using a JumpStart profile.

`ok boot net - install http://131.141.2.32/jumpstart/config.tar`

For a description of the boot command options, see the following table.

---

**SPARC only** – The system checks hardware and system components and your SPARC based system boots. Booting lasts several minutes.

4. If you did not preconfigure system information in the sysidcf file, when prompted, answer the questions about system configuration.

Note – Starting with the Oracle Solaris 10 9/10 release, if you do not include the auto_reg keyword in the sysidcf file, you will be prompted to provide your support credentials and proxy information for Auto Registration.

5. Follow the instructions on the screen to install the software.

When the JumpStart program finishes installing the Oracle 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`

---

**SPARC: Command Reference for the boot Command**

The syntax of the boot command is as follows:

`ok boot [cd-dvd|net] - install [url|ask] options`

The following table describes the command-line options for the boot command that are appropriate for a JumpStart installation.
Option Description

[cd–dvd|net] Specifies to boot from a CD or a DVD or to boot from an install server on the network.
- cd–dvd - Use cdrom to boot from a CD or a DVD.
- net - Specifies to boot from an install server on the network.

[url|ask] Specifies the location of the JumpStart files or prompts you for the location.
- url - Specifies the path to the files. You can specify a URL for files that are located in an HTTP or HTTPS server:
  HTTP server
  http://server-name:IP-address/jumpstart-dir/
  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. The prompt happens after the system boots and connects to the network. If you use
  this option, you are not able to do a completely hands-off JumpStart installation.
  If you bypass the prompt by pressing Return, the Oracle Solaris installation program interactively
  configures the network parameters. The installation program then prompts you for the location of the
  compressed configuration file.

---

**x86: Setting Up a System for a JumpStart Installation (Task Map)**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine whether 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 Oracle Solaris 10 8/11 software. This decision determines how to specify the <code>fdisk(1M)</code> keyword in the system's profile.</td>
<td>“x86: fdisk Profile Keyword (UFS and ZFS)” on page 116</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
<td>For Instructions</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Check whether the system is supported.</td>
<td>Check the hardware documentation for system support in the Oracle Solaris environment.</td>
<td>Hardware manufacturer's documentation</td>
</tr>
<tr>
<td>Check whether the system has enough disk space for the Oracle Solaris software.</td>
<td>Verify that you have planned enough space to install the Oracle Solaris software on your system.</td>
<td>Chapter 3, &quot;System Requirements, Guidelines, and Upgrade Information,&quot; in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade</td>
</tr>
<tr>
<td>(Optional) Set system parameters.</td>
<td>You can preconfigure system information to avoid being prompted for the information during the installation or upgrade.</td>
<td>Chapter 2, &quot;Preconfiguring System Configuration Information (Tasks),&quot; in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations</td>
</tr>
<tr>
<td>Prepare the system for JumpStart installation.</td>
<td>Create and validate a rules file and profile files.</td>
<td>Chapter 3, &quot;Preparing JumpStart Installations (Tasks)&quot;</td>
</tr>
<tr>
<td>(Optional) Prepare optional JumpStart features.</td>
<td>If you are using begin scripts, finish scripts, or other optional features, prepare the scripts or files.</td>
<td>Chapter 4, &quot;Using Optional JumpStart Features (Tasks),&quot; and Chapter 5, &quot;Creating Custom Rule and Probe Keywords (Tasks)&quot;</td>
</tr>
<tr>
<td>(Optional) Prepare to install the Oracle Solaris software from the network.</td>
<td>To install a system from a remote Oracle Solaris Operating System for x86 Platforms DVD or Oracle Solaris Software For x86 Platforms CD image, you need to set up the system to boot and install from an install server or a boot server.</td>
<td>Chapter 6, &quot;Installing From the Network With CD Media (Tasks),&quot; in Oracle Solaris 10 1/13 Installation Guide: Network-Based Installations</td>
</tr>
<tr>
<td>(Optional) Prepare for a Flash Archive installation.</td>
<td>Set up specifics for a Flash Archive installation.</td>
<td>&quot;To Prepare to Install a Flash Archive With a JumpStart Installation&quot; on page 77</td>
</tr>
<tr>
<td>Perform an installation or upgrade.</td>
<td>Boot the system to initiate the installation or upgrade.</td>
<td>&quot;x86: How to Perform an Installation or Upgrade With the JumpStart Program and GRUB&quot; on page 82</td>
</tr>
</tbody>
</table>
x86: Performing a JumpStart Installation

During a 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 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 rules.ok file and begins to install the system, based on the matched rule’s profile.

You can install a flash archive with JumpStart. For instructions, see “To Prepare to Install a Flash Archive With a JumpStart Installation” on page 77.

Choose the following procedures to perform an installation or upgrade with the JumpStart program and GRUB.

- For a standard JumpStart procedure, see “x86: How to Perform an Installation or Upgrade With the JumpStart Program and GRUB” on page 82.
- To perform a JumpStart by editing the GRUB command, see “x86: How to Perform an Installation or Upgrade With the JumpStart Program and GRUB” on page 82.

▼ x86: How to Perform an Installation or Upgrade With the JumpStart Program and GRUB

Use this procedure to install the Oracle Solaris OS for an x86 based system with the GRUB.

**Before You Begin**

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

1. **If you are using a profile diskette, insert the profile diskette into the system's diskette drive.**
   A profile diskette is no longer used to boot the system but, a diskette can be prepared that includes only the JumpStart directory. The diskette can then be used situations such as performing a JumpStart installation and booting off the CD-ROM.

2. **Boot the system.**

   - **If you boot from the Oracle Solaris Operating System DVD or the Oracle Solaris Software - 1 CD, insert the disc.**
     
     Your system's BIOS must support booting from a DVD or CD. If you are booting from a DVD or CD, change the boot setting in your system's BIOS and set to boot from DVD or CD media. For more information, See your hardware documentation.
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.

3 If the system is off, turn the system on. If the system is on, reboot the system.

The GRUB menu is displayed. This menu provides a list of boot entries.

GNU GRUB version 0.95 (631K lower / 2095488K upper memory)

+-------------------------------------------------------------------+
| Oracle Solaris 10 8/11  image_directory | |
| Solaris Serial Console ttya | |
| Solaris Serial Console ttyb (for lx50, v60x and v65x | |
+-------------------------------------------------------------------+

Use the ^ and v keys to select which entry is highlighted. Press enter to boot the selected OS, ‘e’ to edit the commands before booting, or ‘c’ for a command-line.

image-directory is the name of the directory where the installation image is located. The path to the JumpStart files was defined with the add_install_client command and the -c option.

4 Determine whether you want to perform a standard boot of the OS or modify the GRUB boot command before performing the JumpStart installation, for example, for debugging purposes.

- To perform a standard boot process, press Enter with Oracle Solaris 10 8/11 highlighted.

- To modify the GRUB boot command and perform a custom boot process:
  
a. Stop the booting process by typing e.
   The GRUB edit menu is displayed.

   kernel /I86PC.Solaris_11-8/multiboot kernel/unix -B console=ttyb,\
   install_media=131.141.2.32:/export/mary/v11 /\n   module /I86PC.Solaris_11-8/x86.new

b. Use the arrow keys to select the boot entry.

c. Type e to access the editor.
   A command that is similar to the following example displays.

   grub edit>kernel /I86PC.Solaris_11-8/multiboot kernel/unix -B \\
   console=ttyb,install_media=131.141.2.32:/export/mary/\ 
   module /I86PC.Solaris_11-8/x86.new

d. Edit the command by typing the options that you need.
   The syntax for a JumpStart installation is the following.

   grub edit>kernel /I86PC.Solaris_11-image_directory/multiboot kernel/unix/ \
   - install [url|ask] options -B install_media=media-type

   For a description of JumpStart options, see “x86: Command Reference for Booting the System” on page 84.
In the following example, the OS is installed over the network with a JumpStart profile:

```
kernel /I86PC.Solaris_11-8/multiboot kernel/unix/ - install \
   -B install_media=131.141.2.32:/export/mary/v11 \
   module /I86PC.Solaris_11-8/x86.new
```

e. **To accept the edits, press Enter.**

   **Note** – To return to the GRUB main menu without saving your changes, press Escape.

   Your changes are saved and the GRUB main menu is displayed.

f. **Type b to begin the boot process.**

   The installation menu is displayed.

5 **Type 2 and press Enter to select the JumpStart method.**

   The JumpStart installation begins.

   **Note** – If you do not make a selection within 30 seconds, the Oracle Solaris interactive installation program begins. You can stop the timer by typing any key at the command line.

6 **If you did not preconfigure system information in the sysidcfg file, when prompted, answer the questions about system configuration.**

   **Note** – Starting with the Oracle Solaris 10 9/10 release, if you do not include the auto_reg keyword in the sysidcfg file, you will be prompted to provide your support credentials and proxy information for Auto Registration.

7 **Follow the instructions on the screen to install the software.**

   When the JumpStart program finishes installing the Oracle Solaris software, the system reboots automatically. Also, the GRUB menu.lst file is automatically updated. Then the instance of Oracle Solaris that you have installed appears in the next use of the GRUB menu.

   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

---

**x86: Command Reference for Booting the System**

The following table describes the command-line options for the GRUB menu boot command. The options listed are appropriate for a JumpStart installation.
The syntax of the boot command is as follows:

```
kernel /I86PC.Solaris_11-image-directory/multiboot kernel/unix/ - install \ [url|ask] options -B install_media=media-type
```

### TABLE 6–4 GRUB Menu Boot Command Reference

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>- install</td>
<td>Performs a JumpStart installation.</td>
</tr>
</tbody>
</table>

In the following example, the system boots from DVD media and the following options were used:

- `- install` performs a JumpStart
- `file://jumpstart/config.tar` finds the JumpStart profile on the local disk

```
kerneld /I86pc.Solaris_11.8/multiboot - install file://jumpstart/config.tar \ -B install_media=dvdrom module /I86Solaris_11.8/x86.new
```
### TABLE 6–4 GRUB Menu Boot Command Reference (Continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[url] ask</td>
<td>Specifies the location of the JumpStart files or prompts you for the location.</td>
</tr>
<tr>
<td>url</td>
<td>Specifies the path to the files. You can specify a URL for files that are located on an HTTP or HTTPS server: The syntax for an HTTP server is the following: <a href="http://server-name:IP-address/jumpstart-dir/">http://server-name:IP-address/jumpstart-dir/</a> compressed-config-file &amp; proxy-info</td>
</tr>
<tr>
<td>■</td>
<td>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:</td>
</tr>
<tr>
<td>kernel /I86pc.Solaris_11.8/multiboot install \ <a href="http://192.168.2.1/jumpstart/config.tar">http://192.168.2.1/jumpstart/config.tar</a> \ -B install_media=192.168.2.1/export/Solaris_11.8/boot \ module /I86PC.Solaris_11.8/x86.new</td>
<td></td>
</tr>
<tr>
<td>■</td>
<td>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:</td>
</tr>
<tr>
<td>kernel /I86pc.Solaris_11.8/multiboot install \ <a href="http://www.shadow.com/jumpstart/config.tar&amp;proxy=131.141.6.151">http://www.shadow.com/jumpstart/config.tar&amp;proxy=131.141.6.151</a> \ -B install_media=192.168.2.1/export/Solaris_11.8/boot \ module /I86PC.Solaris_11.8/x86.new</td>
<td></td>
</tr>
<tr>
<td>■ ask</td>
<td>Specifies that the installation program prompt you to type the location of the compressed configuration file. You are prompted after the system boots and connects to the network. If you use this option, you are not able to do a completely hands-off JumpStart installation.</td>
</tr>
<tr>
<td>If you bypass the prompt by pressing Return, the Oracle Solaris installation program interactively configures the network parameters. The installation program then prompts you for the location of the compressed configuration file.</td>
<td></td>
</tr>
<tr>
<td>The following example performs a JumpStart installation and boots from DVD media. You are prompted to type the location of the configuration file after the system connects to the network.</td>
<td></td>
</tr>
<tr>
<td>kernel /boot/multiboot kernel/unix install ask -B \ install_media=192.168.2.1:export/sol_11_x86/boot module \ /I86PC.Solaris_11.8</td>
<td></td>
</tr>
</tbody>
</table>
This chapter provides an example of setting up and installing Oracle Solaris software on both SPARC based and x86 based systems by using a JumpStart installation.

**Note** – If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, “Installing a ZFS Root Pool With JumpStart,” for limitations and profile examples.

This chapter covers the following topics:

- “Sample Site Setup” on page 87
- “Create an Install Server” on page 89
- “x86: Create a Boot Server for Marketing Systems” on page 90
- “Create a JumpStart Directory” on page 91
- “Share the JumpStart Directory” on page 91
- “SPARC: Create the Engineering Group’s Profile” on page 91
- “x86: Create the Marketing Group’s Profile” on page 92
- “Update the rules File” on page 92
- “Validate the rules File” on page 93
- “SPARC: Set Up Engineering Systems to Install From the Network” on page 93
- “x86: Set Up Marketing Systems to Install From the Network” on page 94
- “SPARC: Boot the Engineering Systems and Install Oracle Solaris Software” on page 95
- “x86: Boot the Marketing Systems and Install Oracle Solaris Software” on page 95

**Sample Site Setup**

The following figure 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 Oracle Solaris 10 8/11 software from the network.
Create an Install Server

Because the groups need to install Oracle Solaris 10 8/11 software from the network, make server-1 an install server for both groups. 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 either of the following media.

- Oracle Solaris Software CDs and the Oracle Solaris Languages CDs

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

- Oracle Solaris Operating System DVD

The image must be copied from the disc to an empty directory, which in these examples are the `sparc_10` directory and the `x86_10` directory.

**EXAMPLE 7-1  SPARC: Copying the Oracle Solaris 10 8/11 CDs**

1. Insert the Oracle Solaris Software for SPARC Platforms - 1 CD in the CD-ROM drive that is attached to server-1 and type the following commands:
   ```
   server-1# mkdir -p /export/install/sparc_10
   server-1# cd /mymountpoint/Solaris_10/Tools
   server-1# ./setup_install_server /export/install/sparc_10
   ```
2. Insert the Oracle Solaris Software for SPARC Platforms - 2 CD in the CD-ROM drive that is attached to server-1 and type the following commands:
   ```
   server-1# cd /mymountpoint/Solaris_10/Tools
   server-1# ./add_to_install_server /export/install/sparc_10
   ```
3. Repeat the previous command for each Oracle Solaris software version you want to install.
4. Insert the first SPARC: Oracle Solaris Languages for SPARC Platforms CD in the CD-ROM drive that is attached to server-1 and type the following commands:
   ```
   server-1# cd /mymountpoint/Solaris_10/Tools
   server-1# ./add_to_install_server /export/install/sparc_10
   ```
5. Repeat the previous command for each SPARC: Oracle Solaris Languages for SPARC Platforms CD.

**EXAMPLE 7-2  SPARC: Copying the Oracle Solaris 10 8/11 DVD**

Insert the Oracle Solaris Operating System for SPARC Platforms DVD in the DVD-ROM drive that is attached to server-1 and type the following commands:
```
server-1# mkdir -p /export/install/sparc_10
server-1# cd /mymountpoint/Solaris_10/Tools
server-1# ./setup_install_server /export/install/sparc_10
```
EXAMPLE 7-3  x86: Copying the Oracle Solaris 108/11 CDs

1. Insert the Oracle Solaris Software for x86 Platforms - 1 CD in the CD-ROM drive that is attached to server-1 and type the following commands:

   ```
   server-1# mkdir -p /export/install/x86_10
   server-1# cd /mymountpoint/Solaris_10/Tools
   server-1# ./setup_install_server /export/install/x86_10
   ```

2. Insert the Oracle Solaris Software for x86 Platforms - 2 CD in the CD-ROM drive that is attached to server-1 and type the following commands:

   ```
   server-1# cd /mymountpoint/Solaris_10/Tools
   server-1# ./add_to_install_server /export/install/x86_10
   ```

3. Repeat the previous command for each Oracle Solaris software version you want to install.

4. Insert the first Oracle Solaris Languages for x86 Platforms CD in the CD-ROM drive that is attached to server-1 and type the following commands:

   ```
   server-1# cd /mymountpoint/Solaris_10/Tools
   server-1# ./add_to_install_server /export/install/x86_10
   ```

5. Repeat the previous command for each Oracle Solaris Languages for x86 Platforms CD.

EXAMPLE 7-4  x86: Copying the Oracle Solaris Operating System for x86 Platforms DVD

Insert the Oracle Solaris Operating System for x86 Platforms DVD in the DVD-ROM drive that is attached to server-1 and type the following commands:

```
server-1# mkdir -p /export/install/x86_10
server-1# cd /mymountpoint/Solaris_10/Tools
server-1# ./setup_install_server /export/install/x86_10
```

---

x86: Create a Boot Server for Marketing Systems

Systems cannot boot from an install server on a different subnet, so server-2 must be a boot server on the marketing group’s subnet. Use the `setup_install_server(1M)` command to copy the boot software from the Oracle Solaris Operating System for x86 Platforms DVD or the Oracle Solaris Software for x86 Platforms - 1 CD. The boot software is copied to the server-2 local disk in the `/export/boot` directory.

Choose the media and install the boot software to local disk. With the CD or DVD inserted into the drive that is attached to server-2, you would issue the following commands:

```
server-2# cd /mymountpoint/Solaris_10/Tools
server-2# ./setup_install_server -b /export/boot
```

The `-b` specifies that `setup_install_server` is to copy the boot information to the `/export/boot` directory.
Create a JumpStart Directory

Now that the install server and boot server are set up, create a JumpStart directory on server-1 using any system on the network. This directory holds files that are required for a JumpStart installation of Oracle Solaris software. Set up this directory by copying the sample directory from the Oracle Solaris Operating System DVD image or from the Oracle Solaris Software - 1 CD image that has been copied to /export/install:

```
server-1# mkdir /jumpstart
server-1# cp -r /export/install/sparc_10/Solaris_10/Misc/jumpstart_sample /jumpstart
```

Share the JumpStart Directory

To make the rules file and profiles accessible to systems on the network, share the /jumpstart directory. To enable the sharing of a directory, add the following line to the /etc/dfs/dfstab file:

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

Then, at the command line, issue the shareall command:

```
server-1# shareall
```

SPARC: Create the Engineering Group's Profile

For the engineering systems, create a file that is named eng_prof in the /jumpstart directory. The eng_prof file contains the following entries, which define the Oracle Solaris 10 8/11 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
```

The previous example profile specifies the following installation information:

```
install_type The installation is to be treated as an initial installation, as opposed to an upgrade.

system_type The engineering systems are stand-alone systems.

partitioning The JumpStart software uses default disk partitioning for installing Oracle Solaris software on the engineering systems.

cluster The Developer System Support software group is to be installed.
```
filesys Each system in the engineering group is to have 512 MB 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 Oracle Solaris 10 8/11 software to be installed on systems in the marketing group:

```
install_type  initial_install
system_type   standalone
partitioning  default
cluster       SUNWCuser
package       SUNWaudio
```

The previous example profile specifies the following installation information:

- **install_type**: The installation is to be treated as an initial installation, as opposed to an upgrade.
- **system_type**: The marketing systems are stand-alone systems.
- **partitioning**: The JumpStart software is to use default disk partitioning for installing Oracle Solaris on the marketing systems.
- **cluster**: The End User Oracle Solaris Software Group is to be installed.
- **package**: The audio demo software package is to be added to each system.

**Update the rules File**

The Oracle Solaris installation program uses the rules in the `rules` file to select the correct installation (profile) for each system during a JumpStart installation.

At this site, each department is located on its own subnet and has its own network address. The engineering department is located on subnet 255.222.43.0. The marketing department is located on 255.222.44.0. Use this information to control how the engineering and marketing systems are installed with the Oracle Solaris 10 8/11 software. In the `/jumpstart` directory, 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 Oracle Solaris 10 8/11 software by using the `eng_prof` profile. The systems on the 255.222.44.0 network are to be installed with the Oracle Solaris 10 8/11 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 Oracle Solaris 10 8/11 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 8–1 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, 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, use the add_install_client command on the install server, server-1, to set up the engineering systems to install the Oracle Solaris software from the install server. server-1 is also the boot server for the engineering group’s subnet.

server-1# cd /export/install/sparc_10/Solaris_10/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 example add_install_client command, the options that are used have the following meanings:

- c Specifies the server (server-1) and path (/jumpstart) to the JumpStart directory. Use this option if you are using NFS.

Note – If you are not using NFS, you specify the path to the JumpStart directory by using the following commands:

- **For SPARC based systems**, specify the path in the boot command
- **For x86 based systems**, specify the path by editing the GRUB menu entry

host-eng1 The name of a system in the engineering group.
host-eng2 The name of another system in the engineering group.
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 Oracle Solaris software from the install server (`server-1`):

```
server-2# cd /marketing/boot-dir/Solaris_10/Tools
server-2# ./add_install_client -s server-1:/export/install/x86_10 \
    -c server-1:/jumpstart host-mkt1 i86pc
server-2# ./add_install_client -s server-1:/export/install/x86_10 \
    -c server-1:/jumpstart host-mkt2 i86pc
server-2# ./add_install_client -d -s server-1:/export/install/x86_10 \
    -c server-1:/jumpstart SUNW.i86pc i86pc
server-2# ./add_install_client -c server-1:/jumpstart host-mkt1 sun4u
server-2# ./add_install_client -c server-1:/jumpstart host-mkt2 sun4u
```

In the `add_install_client` command, the options that are used have the following meanings:
- `-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.
- `-s` Specifies the install server (`server-1`) and the path to the Oracle Solaris software (`/export/install/x86_10`).
- `-c` Specifies the server (`server-1`) and path (`/jumpstart`) to the JumpStart directory. Use this option if you are using NFS.

Note – If you are not using NFS, you specify the path to the JumpStart directory by using the following commands:
- For **SPARC based systems**, specify the path in the boot command
- For **x86 based systems**, specify the path by editing the GRUB menu entry

- `host-mkt1` The name of a system in the marketing group.
- `host-mkt2` The name of another system in the marketing 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.
The DHCP class name for all Oracle Solaris x86 clients. If you want to configure all Oracle Solaris x86 DHCP clients with a single command, use this class name.

Specifies the platform group of the systems that use this boot server. The platform name represents x86 based systems.

### SPARC: Boot the Engineering Systems and Install Oracle Solaris Software

After setting up the servers and files, boot the engineering systems by using the following `boot` command at the `ok` (PROM) prompt of each system:

```
ok boot net - install
```

The Oracle Solaris OS is automatically installed on the engineering group’s systems.

### x86: Boot the Marketing Systems and Install Oracle Solaris Software

You can boot the system from one of the following:

- Oracle Solaris Software for x86 Platforms - 1 CD
- Oracle Solaris Operating System for x86 Platforms DVD
- The network by using PXE network boot

Oracle Solaris software is automatically installed on the marketing group’s systems.
This chapter lists keywords and values that you can use in the rules file, profiles, and begin and finish scripts.

**Note** - If you are installing an Oracle Solaris ZFS root pool, see Chapter 9, "Installing a ZFS Root Pool With JumpStart," for limitations and profile examples. Also, for a list of ZFS-specific keywords and keywords that can be used within a profile, see Table 8–2.

This chapter covers the following topics:
- "Rule Keywords and Values" on page 97
- "Profile Keywords and Values" on page 101
- "JumpStart Environment Variables" on page 145
- "Probe Keywords and Values" on page 147

## Rule Keywords and Values

The following table 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 28.

<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 as follows:</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: 1386</td>
<td></td>
</tr>
</tbody>
</table>

8  CHAPTER 8

JumpStart Keyword Reference

97
### Rule Keywords and Values

**Keyword** | **Value** | **Matches**
--- | --- | ---
`disksize` | `disk-name size-range` | The name and size of a system's disk in MB. For example:

- `disksize c0t3d0 250-300`

  In the example, the JumpStart program attempts to match a system disk that is named `c0t3d0`. The disk can hold between 250 and 300 MB of information.

- `disksize rootdisk 750-1000`

  In the example, the JumpStart program attempts to match a disk in the following order:
  1. A system disk that contains a preinstalled boot image
  2. The `c0t3d0s0` disk, if the disk exists
  3. The first available disk that can hold between 750 MB and 1 GB of information

  Note: When calculating size-range, remember that a MB equals 1,048,576 bytes. A disk might be advertised as a 535 MB disk, but the disk might contain only 510 million bytes of disk space. The JumpStart program views the 535 MB disk as a 510 MB disk because $535,000,000 / 1,048,576 = 510$. A 535 MB disk does not match a size-range equal to 530–550.

`domainname` | `domain-name` | A system's domain name, which controls how a naming service determines information.

If you have a system already installed, the `domainname` command reports the system's domain name.

`hostaddress` | `IP-address` | A system's IP address.

`hostname` | `host-name` | A system's host name.

If you have a system that is already installed, the `uname -n` command reports the system's host name.
### TABLE 8–1 Descriptions of Rule Keywords and Values  
(Continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>installed</strong></td>
<td><strong>slice version</strong></td>
<td>A disk that has a root (/) file system that corresponds to a particular version of Oracle Solaris software.</td>
</tr>
<tr>
<td></td>
<td>slice – A disk slice name in the form cwtxdysz, 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:</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><strong>version</strong> – A version name or the special words any or upgrade. If any is used, any Oracle Solaris release is matched. If upgrade is used, any Oracle Solaris release that is supported and can be upgraded is matched.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the JumpStart program finds an Oracle Solaris release but is unable to determine the version, the version that is returned is SystemV.</td>
<td></td>
</tr>
<tr>
<td><strong>karch</strong></td>
<td><strong>platform-group</strong></td>
<td>A system's platform group.</td>
</tr>
<tr>
<td></td>
<td>Valid values are sun4u, i86pc, and prep. For a list of systems and their corresponding platform groups, see Oracle Solaris Sun Hardware Platform Guide at <a href="http://www.oracle.com/technetwork/indexes/documentation/index.html">http://www.oracle.com/technetwork/indexes/documentation/index.html</a>.</td>
<td></td>
</tr>
<tr>
<td><strong>memsize</strong></td>
<td><strong>physical-mem</strong></td>
<td>A system's physical memory size in MB.</td>
</tr>
<tr>
<td></td>
<td>The value must be a range of MB, x–x, or a single MB value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: memsize 64-128</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The example tries to match a system with a physical memory size between 64 and 128 MB.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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>
<td></td>
</tr>
</tbody>
</table>
**TABLE 8-1  Descriptions of Rule Keywords and Values  (Continued)**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Matches</th>
</tr>
</thead>
</table>
| model   | platform-name | A system's platform name. See the Oracle Solaris Sun Hardware Platform Guide at http://www.oracle.com/technetwork/indexes/documentation/index.html for a list of valid platform names. To find the platform name of an installed system, use the `uname -i` command or the output of the `prtconf` command, line 5.  
  **Note** – If `platform-name` contains spaces, you must replace spaces with underscores (_).  
  For example:  
  SUNW, Sun_4_50 |
| network | network-num | A system's network number, which the JumpStart program determines by performing a logical AND between the system's IP address and the subnet mask.  
  Example:  
  `network 192.168.2.0`  
  The example tries to match a system with a 192.168.2.8 IP address, if the subnet mask is 255.255.255.0. |
| osname  | Solaris_x  | A version of Oracle Solaris software that is already installed on a system.  
  For example:  
  `osname Solaris 10`  
  In the example, the JumpStart program attempts to match a system with the Oracle Solaris 10 8/11 OS already installed. |
### Table 8-1 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>probe-keyword</td>
<td>A valid probe keyword or a valid custom probe keyword.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example: probe disks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The example returns the size of a system’s disks in MB and in kernel probe order, for example,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>c0t3d0s1, c0t4d0s0, on a SPARC based system. The JumpStart program sets the SI_DISKLIST,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SI_DISKSIZES, SI_NUMDISKS, and SI_TOTALDISK environment variables.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note – The probe keyword is unique in that the keyword does not attempt to match an attribute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and run a profile. The probe keyword returns a value. Consequently, you cannot specify begin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scripts, profiles, and finish scripts with the probe rule keyword.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Probe keywords are described in Chapter 5, “Creating Custom Rule and Probe Keywords (Tasks).”</td>
</tr>
<tr>
<td>totaldisk</td>
<td>size-range</td>
<td>The total disk space on a system in MB. The total disk space includes all the operational disks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>that are attached to a system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note – When calculating size-range, remember that one MB equals 1,048,576 bytes. A disk might be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>advertised as a 535 MB disk, but the disk might have only 510 million bytes of disk space. The</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JumpStart program views the 535 MB disk as a 510 MB disk because 535,000,000 / 1,048,576 = 510.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A 535 MB disk does not match a size-range equal to 530–550.</td>
</tr>
</tbody>
</table>

---

**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 32. These keywords are for installing UFS and ZFS file systems. If the keyword can be used in a ZFS profile, the term “ZFS” is noted.
Profile Keywords Quick Reference

The following table 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. Also, these keywords are for a UFS file system unless noted that the keyword can be used in a ZFS root pool profile.

<table>
<thead>
<tr>
<th>Profile Keywords and Values</th>
</tr>
</thead>
</table>

### TABLE 8-2 Profile Keywords Overview

<table>
<thead>
<tr>
<th>Profile Keyword</th>
<th>Stand-alone System (Nonnetworked)</th>
<th>Stand-alone System (Networked) or Server</th>
<th>OS Server</th>
<th>Upgrade</th>
<th>Flash Archive</th>
<th>Flash Archive Differential Archive</th>
<th>Upgrade With Disk Space Reallocation</th>
<th>Both UFS and ZFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>archive_location</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>backup_media</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>boot_device (UFS and ZFS)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>bootenv (UFS and ZFS)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>client_arch</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>client_root</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>client_swap</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster (adding software groups)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>cluster (adding or deleting clusters)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>dontuse</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>fdisk (x86 only)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>filesystems (mounting remote file systems)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>filesystems (creating local file systems)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>filesystems (creating mirrored file systems)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>forced_deployment</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>geo</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>install_type</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
### Profile Keywords Overview (Continued)

<table>
<thead>
<tr>
<th>Profile Keyword</th>
<th>Stand-alone System (Nonnetworked)</th>
<th>Stand-alone System (Networked) or Server</th>
<th>OS Server</th>
<th>Upgrade</th>
<th>Flash Archive</th>
<th>Flash Archive Differential Archive</th>
<th>Upgrade With Disk Space Reallocation</th>
<th>Both UFS and ZFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>layout_constraint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>local_customization</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>locale</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>metab (creating state database replicas)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>no_master_check</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>no_content_check</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>num_clients</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>package</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>partitioning</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>patch</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>pool</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X (ZFS root pools only)</td>
</tr>
<tr>
<td>root_device</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>system_type</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>usedisk</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>iscsi_target_name</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>iscsi_target_ip</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>iscsi_target_lun</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>iscsi_target_port</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>iscsi_target_slice</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

### Profile Keyword Descriptions and Examples

#### archive_location Keyword

`archive_location retrieval-type location`
retrieval-type

The values of retrieval-type and location depend on where the 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:

- “Archive Stored on an NFS Server” on page 104
- “Archive Stored on an HTTP or HTTPS Server” on page 105
- “Archive Stored on an FTP Server” on page 106
- “Archive Stored on a Local Tape” on page 107
- “Archive Stored on a Local Device” on page 108
- “Archive Stored on a Local File” on page 109

location

Specifics for locations are not in the following sections.

Caution – A flash archive cannot be properly created when a non-global zone is installed. The Flash Archive feature is not compatible with the Oracle Solaris Zones partitioning technology. If you create a flash archive, the resulting archive is not installed properly when the archive is deployed under these conditions:

- The archive is created in a non-global zone
- The archive is created in a global zone that has non-global zones installed

Archive Stored on an NFS Server

If the archive is stored on an NFS server, use the following syntax for the archive_location keyword.

archive_location nfs server-name:/path/filename retry n

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 Flash Archive installation utilities replace $HOST with the name of the clone system that you are installing.

filename The name of the flash archive file.

retry n An optional keyword. n is the maximum number of times the Flash Archive utilities attempt to mount the archive.

EXAMPLE 8–1 Archive Stored on an NFS Server

archive_location nfs golden:/archives/usrarchive

archive_location nfs://golden/archives/usrarchive
Archive Stored on an HTTP or HTTPS 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
```

If the archive is stored on an HTTPS server, use the following syntax for the `archive_location` keyword.

```
archive_location https://server-name:port/path/filename optional-keywords
```

- `server-name` The name of the server where you stored the archive.
- `port` An optional port. `port` can be a port number or the name of a TCP service that has a port number that is determined at runtime.
  - If you do not specify a port, the Flash Archive 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 `$HOST`, the Flash Archive installation utilities replace `$HOST` with the name of the clone system that you are installing.
- `filename` The name of the flash archive file.
- `optional-keywords` The optional keywords that you can specify when you retrieve a flash archive from an HTTP server.

### TABLE 8–3 Optional Keywords to Use With `archive_location HTTP`

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value Definition</th>
</tr>
</thead>
</table>
| `auth basic user-name password` | 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.  
  **Note** – The use of this authentication method in a profile that is intended for use with JumpStart is risky. Unauthorized users might have access to the profile file that contains the password. |
Optional Keywords to Use With archive_location HTTP

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout min</td>
<td>The <code>timeout</code> keyword enables you to specify, in minutes, the maximum length of time that is allowed to pass without receipt of data from the HTTP server. If a timeout occurs, the connection is closed, reopened, and resumed. If you specify a <code>timeout</code> value of 0 (zero), the connection is not reopened.</td>
</tr>
<tr>
<td></td>
<td>• If a timeout reconnection occurs, the Flash Archive installation utilities attempt to resume the installation at the last known position in the archive. If the Flash Archive 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></td>
<td>• If a timeout reconnection occurs while a package is being installed, the package is retried from the beginning of the package and the data that was retrieved prior to the timeout is discarded.</td>
</tr>
<tr>
<td>proxy host:port</td>
<td>The <code>proxy</code> keyword enables you to specify a proxy host and proxy port. You can use a proxy host to retrieve a flash archive from the other side of a firewall. You must supply a proxy port when you specify the <code>proxy</code> keyword.</td>
</tr>
</tbody>
</table>

Example 8–2 Archive Stored on a HTTP or HTTPS Server

```
archive_location http://silver/archives/usrarchive.flar timeout 5
```

Example of the `auth basic username password` keyword:

```
archive_location http://silver/archives/usrarchive.flar timeout 5 user1 secret
```

**Archive Stored on an FTP Server**

If the archive is stored on an FTP server, use the following syntax for the `archive_location` keyword.

```
archive_location ftp://username:password@server-name:port/path/filename optional-keywords
```

- `username: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.
- `port` An optional port. `port` can be a port number or the name of a TCP service that has a port number that is determined at runtime.
  
  If you do not specify a port, the Flash Archive 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 `$HOST`, the Flash Archive installation utilities replace `$HOST` with the name of the clone system that you are installing.
filename The name of the flash archive file.

optional-keywords The optional keywords that you can specify when you retrieve a flash archive from an FTP server.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Definition</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 HTTP server. If a timeout occurs, the connection is closed, reopened, and resumed. If you specify a timeout value of 0 (zero), the connection is not reopened.  
  ■ If a timeout reconnection occurs, the Flash Archive installation utilities attempt to resume the installation at the last known position in the archive. If the Flash Archive 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.  
  ■ If a timeout reconnection occurs while a package is being installed, the package is retried from the beginning of the package and the data that was retrieved prior to the timeout is discarded. |
| proxy host:port |       | The proxy keyword enables you to specify a proxy host and proxy port. You can use a proxy host to retrieve a flash archive from the other side of a firewall. You must supply a proxy port when you specify the proxy keyword. |

EXAMPLE 8–3 Archive Stored on an FTP Server

archive_location ftp://user1:secret@silver/archives/usrarchive.flar timeout 5

Archive Stored on a 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 flash archive. If the device name is a canonical path, the Flash Archive 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 Flash Archive 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 Flash Archive installation utilities retrieve the archive from the current position on the tape drive. By specifying position, you can place a begin script or a syslogcfg file on the tape drive before the archive.
EXAMPLE 8-4  Archive Stored on a Local Tape

archive_location local_tape /dev/rmt/0n 5

archive_location local_tape 0n 5

Archive Stored on a Local Device

You can retrieve a flash archive from a local device if you stored the flash archive on a file system-oriented, random-access device, such as a diskette or a DVD. 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 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 Flash Archive installation utilities add /dev/dsk/ to the path.

path The path to the flash archive, relative to the root of the file system on the device you specified. If the path contains $HOST, the Flash Archive installation utilities replace $HOST with the name of the clone system that you are installing.

filename The name of the 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 Flash Archive installation utilities attempt to mount a UFS file system. If the UFS mount fails, the Flash Archive installation utilities attempt to mount an HSFS file system.

EXAMPLE 8–5  Archive Stored on a Local Device

To retrieve an archive from a local hard drive that is formatted as a UFS file system:

archive_location local_device c0t0d0s0 /archives/$HOST

To retrieve an archive from a local CD-ROM that has an HSFS file system:

archive_location local_device c0t0d0s0 /archives/usrarchive
Archive Stored on a 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 JumpStart installation, you boot the system from a DVD, CD, or an NFS-based miniroot. The installation software is loaded and run from this miniroot. Therefore, a flash archive that you stored in the DVD, CD, or NFS-based miniroot is accessible as a local file. Use the following syntax for the archive_location keyword.

```
archive_location local_file path/filename
```

- `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 Oracle Solaris Software - 1 CD or from the Oracle Solaris Operating System DVD. The system cannot access /net or any other automounted directory when it is booted from the Oracle Solaris Software - 1 CD or from the Oracle Solaris Operating System DVD.
- `filename` The name of the flash archive file.

**EXAMPLE 8–6** Archive Stored on a Local File

```
archive_location local_file /archives/usrarchive
```

backup_media Profile Keyword

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.

<table>
<thead>
<tr>
<th>Valid type Value</th>
<th>Valid path Value</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>local_tape</code></td>
<td><code>/dev/rmt/n</code></td>
<td>A local tape drive on the system that is being upgraded. <code>path</code> must be the character (raw) device path for the tape drive. <code>n</code> is the number of the tape drive.</td>
</tr>
<tr>
<td><code>local_diskette</code></td>
<td><code>/dev/rdiskette</code></td>
<td>A local diskette drive on the system that is being upgraded. <code>path</code> must be the character (raw) device path for the diskette drive. <code>n</code> is the number of the diskette drive. Diskettes that you use for the backup must be formatted.</td>
</tr>
</tbody>
</table>
Profile Keywords and Values

<table>
<thead>
<tr>
<th>Valid <code>type</code> Value</th>
<th>Valid <code>path</code> Value</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>local_filesystem</td>
<td>/dev/dsk/c0t3d0s4</td>
<td>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. <code>path</code> can be a block device path for a disk slice. For example, the tx in /dev/dsk/c0t3d0s4 might not be needed. Or, <code>path</code> can be the absolute path to a file system that is mounted by the <code>/etc/vfstab</code> file.</td>
</tr>
<tr>
<td>remote_filesystem</td>
<td>host:/export</td>
<td>An NFS file system on a remote system. <code>path</code> must include the name or IP address of the remote system, <code>host</code>, and the absolute path to the NFS file system, <code>file-system</code>. The NFS file system must have read/write access.</td>
</tr>
<tr>
<td>remote_system</td>
<td>user@host:/export</td>
<td>A directory on a remote system that can be reached by a remote shell, <code>rsh</code>. The system that is being upgraded must have access to the remote system through the remote system’s <code>.rhosts</code> file. <code>path</code> must include the name of the remote system <code>host</code> and the absolute path to the directory <code>directory</code>. If a user login ID <code>user</code> is not specified, <code>root</code> is used by default.</td>
</tr>
</tbody>
</table>

**EXAMPLE 8–7** backup_media Profile Keyword

- 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 (UFS and ZFS)**

**Note** – The `boot_device` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations. For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.

- 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. `boot_device` must match any `filesys` keywords that specify the root (/) file system and the `root_device` keyword.
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**: `cwt dysz` or `cx dysz`
  - The disk slice where the JumpStart program places the root (`/`) file system, for example, `c0t0d0s0`.
- **x86**: `cwt dys` or `cx dys`
  - 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.

The `eeprom` value enables 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.

**Note** – 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.

**EXAMPLE 8-8**  boot_device Profile Keyword

```
boot_device c0t0d0s2 update
```
**bootenv Profile Keyword (UFS and ZFS)**

**Note** – The `bootenv` keyword can be used for either a UFS file system or a ZFS root pool installation. The usage is different for a ZFS installation.

- For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2
- For a description of how the `bootenv` keyword can be used when installing a ZFS root pool, see "JumpStart Keywords for a ZFS Root (/) File System (Reference)" on page 151

`bootenv createbe bename new-BE-name filesystem mountpoint:device:fs-options [filesystem...]`

bootenv createbe keyword enables you to quickly create an empty and inactive boot environment at the same time you are installing the Oracle Solaris OS. At a minimum, you must create the root (/) file system. The slices are reserved for the file systems specified, but no file systems are copied. The boot environment is named, but not actually created until installed with a flash archive. When the empty boot environment is installed with an archive, file systems are installed on the reserved slices. The following lists the values for `bename` and `filesystem`.

**bename new-BE-name**

- `bename` specifies the name of the new boot environment to be created. `new-BE-name` can be no longer than 30 characters, can contain only alphanumeric characters, and can contain no multibyte characters. The name must be unique on the system.

**filesystem mountpoint:device:fs-options**

- `filesystem` determines the type and number of file systems that are to be created in the new boot environment. At least one slice that contains the root (/) file system must be defined. File systems can be on the same disk or spread across multiple disks.
  - `mountpoint` can be any valid mount point or – (hyphen), indicating a swap slice.
  - `device` must be available when the operating system that is being installed is first booted. The device has no relation to JumpStart special storage devices such as `free`. The device cannot be a Solaris Volume Manager volume or Veritas Volume Manager volume. `device` is the name of a disk device, of the form `/dev/dsk/cwtxdysz`.
  - `fs-options` can be one of the following:
    - `ufs`, which indicates a UFS file system.
    - `swap`, which indicates a swap file system. The swap mount point must be a – (hyphen).

For a profile example and background about using this keyword, see the following references:

- For an example of a profile, see Example 3–11.
For background about using a flash archive, see Chapter 1, "Flash Archive Overview," in Oracle Solaris 10 1/13 Installation Guide: Flash Archives (Creation and Installation).

**client_arch Profile Keyword**

`client_arch karch-value ...

`client_arch` specifies that the operating system server is to support a different platform group than the server uses. If you do not specify `client_arch` in the profile, any diskless client that uses the operating system server must contain the same platform group as the server. You must specify each platform group that you want the operating system server to support.

Valid values for `karch-value` are `sun4u` and `i86pc`. For a detailed list of platform names and various systems, see Oracle Solaris Sun Hardware Platform Guide at [http://www.oracle.com/technetwork/indexes/documentation/index.html](http://www.oracle.com/technetwork/indexes/documentation/index.html).

**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 MB, to allocate for each client. If you do not specify `client_root` in a server’s profile, the installation software allocates 15 MB 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 MB, to allocate for each diskless client. If you do not specify `client_swap` in the profile, 32 MB of swap space is allocated by default.

**Note** – You can use `client_swap` only when `system_type` is specified as `server`.

**EXAMPLE 8–9 client_swap Profile Keyword**
The following example specifies that each diskless client is to have a swap space of 64 MB.

`client_swap 64`
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. The following table shows how the size of swap is determined during a JumpStart installation.

<table>
<thead>
<tr>
<th>Physical Memory (in MB)</th>
<th>Swap Space (in MB)</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. The allocation is different if 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 the table.

**Note** – Physical memory plus swap space must total a minimum of 32 MB.

**cluster Profile Keyword (Adding Software Groups) (UFS and ZFS)**

The cluster keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations.

- For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2
- For a description of how the cluster keyword can be used in a profile for an installation of a ZFS root pool, see "JumpStart Profile Examples for a ZFS Root Pool" on page 154

**cluster group-name**

cluster designates the software group to add to the system.

A software group is a metacluster that contains a collection of clusters and packages. The software group is installed by using the cluster keyword and group-name variable. This cluster keyword can only be installed in an initial installation. This cluster keyword refers to metaclusters found in the clustertoc(4) file.

A cluster is a collection of packages that is named SUNWname. A cluster is installed by using the cluster keyword and cluster-name variable. A cluster can be added or removed from a software group (metacluster) in an initial install or an upgrade.

The group-name for each software group is listed in the following table.
The following limitations apply:

- 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.

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

### cluster Profile Keyword (Adding or Deleting Clusters) (UFS and ZFS)

```
cluster  cluster-name  [add | delete]
```

**Note** – The `cluster` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations.

- For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2
- For a description of how the `cluster` keyword can be used in a profile for an installation of a ZFS root pool, see "JumpStart Profile Examples for a ZFS Root Pool" on page 154

`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**  The name of the cluster that must be in the form SUNWname.

**[add | delete]**  An optional keyword that indicates whether to add or delete the cluster that is specified. Use the value add or delete. If you do not specify add or delete, add is used by default.

When you use `cluster` 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.

**Note** – A software group is a metacluster that contains a collection of clusters and packages. The software group is installed by using the `cluster` keyword and `group-name` variable. This cluster keyword can only be installed in an initial installation. This `cluster` keyword refers to metaclusters found in the `clusters toc(4)` file.

A cluster is collection of packages. Clusters can be grouped together to form a software group (metacluster). A cluster name is always in the form of `SUNWname`. A cluster is installed by using the `cluster` keyword and `cluster-name` variable. A cluster can be added or removed from a software group (metacluster) in an initial install or an upgrade.

dontuse Profile Keyword (UFS and ZFS)

**Note** – The `dontuse` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations.

For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.

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 `cxydorzcydz`, for example, `c0t0d0`.

**Note** – You cannot specify the `dontuse` keyword and the `usedisk` keyword in the same profile.

x86: fdisk Profile Keyword (UFS and ZFS)

**Note** – The `fdisk` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations. For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.

`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.
- An Oracle Solaris fdisk partition that contains a root (/) file system is always designated as the active partition on the disk.

**Note** – 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:

- cxyz or cxyz – A specific disk, for example, c0t3d0.
- rootdisk – The variable that contains the value of the system’s root disk, which is where the installation takes place. The root disk is determined by the JumpStart program as described in “How the System’s Root Disk Is Determined” on page 143.
- 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 – An Oracle 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.

**Note** – You can specify this value only if size is delete.
0xHH – A hexadecimal fdisk partition. HH is a hexadecimal number between 01 and FF.

**Note** – 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>DOS512</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>PCIXOS</td>
<td>2</td>
<td>02</td>
</tr>
<tr>
<td>DOS516</td>
<td>4</td>
<td>04</td>
</tr>
<tr>
<td>EXT505</td>
<td>5</td>
<td>05</td>
</tr>
<tr>
<td>DOSUGE</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>

Use one of the following values:

- **DDD** – An fdisk partition of size DDD in MB 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 the 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) (UFS and ZFS)

Note – The filesys keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations. For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.

filesys server:path server-address mount-point 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.

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 naming 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 naming service that is running on the network, you do not need to specify the server’s IP address.

mount-point 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-point.

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).
EXAMPLE 8–10  filesys Profile Keyword

filesys  sherlock:/export/home/user2 - /home

**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.

- **cwxtdysz** or **cwdysz**
  
  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 143. The *sn* suffix indicates a specific slice on the disk.

  **Note** – The root disk is determined by the JumpStart program and determines where the OS is to be installed. The rules file uses a probe keyword `rootdisk`, but this keyword is used differently than the ”rootdisk” keyword used in the JumpStart profile. You cannot set the place of installation by using the probe keyword `rootdisk` in the rules file. The probe keyword, `rootdisk`, determines where to boot from during the installation. See Table 8–10.

### size

Use one of the following values:
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>num</strong></td>
<td>The size of the file system is set to num in MB.</td>
</tr>
<tr>
<td><strong>existing</strong></td>
<td>The current size of the existing file system is used.</td>
</tr>
<tr>
<td><strong>auto</strong></td>
<td>The size of the file system is automatically determined, depending on the software that is selected.</td>
</tr>
<tr>
<td><strong>all</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>free</strong></td>
<td>The remaining unused space on the disk is used for the file system.</td>
</tr>
<tr>
<td><strong>start:size</strong></td>
<td>The file system is explicitly partitioned. start is the cylinder where the slice begins. size is the number of cylinders for the slice.</td>
</tr>
<tr>
<td><strong>file-system</strong></td>
<td>The file-system value is optional and used when slice is specified as any or cwtxdysz. If file-system is not specified, unnamed is set by default. If unnamed is set, you cannot specify the optional-parameters value. Use one of the following values:</td>
</tr>
<tr>
<td><strong>mount-point</strong></td>
<td>The file system's mount-point name, for example, /var.</td>
</tr>
<tr>
<td><strong>swap</strong></td>
<td>The specified slice is used as swap.</td>
</tr>
<tr>
<td><strong>overlap</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

**Note** - When you use the existing value, you can change the name of an existing slice by specifying file-system as a different mount-point.

**Note** - If free is used as the value to filesys, the filesys entry must be the last entry in a profile.

**Note** - You can specify overlap only when size is existing, all, or start:size.
The specified slice is defined as a raw slice, so slice does not have a mount-point name. If you do not specify file-system, unnamed is used by default.

ignore

The specified slice is not used or recognized by the JumpStart program. You can use this option to specify that you want a file system to be ignored on a disk during installation. The JumpStart program creates a new file system on the same disk with the same name. You can use ignore only when partitioning existing is specified.

optional-parameters

Use one of the following values:

preserve

The file system on the specified slice is preserved.

Note – preserve can be specified only when size is existing and slice is cwtxdysz.

mount-options

One or more mount options, which is the same as the -o option of the mount(1M) command. The mount options are added to the /etc/vfstab entry for the specified mount-point.

Note – If you need to specify more than one mount option, the mount options must be separated by commas and no space (ro, quota, for example).

filesys Profile Keyword (Creating RAID-1 Volumes)

filesys mirror[:name]slice [slice] size file-system optional-parameters

By using the filesys mirror keywords with the values that are listed, the JumpStart program creates the RAID-1 and RAID-0 volumes that are necessary to create a mirrored file system. You can specify filesys mirror more than once to create RAID-1 volumes (mirrors) for different file systems.

Note – The filesys mirror keyword is only supported for initial installations.

name

This optional keyword enables you to name the RAID-1 volume (mirror). Mirror names must start with the letter “d” followed by a number between 0 and 127, for example, d100. If you do not specify a
mirror name, the JumpStart program assigns a mirror name for you. For guidelines about how to name mirrors, see "RAID Volume Name Requirements and Guidelines" in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.

**slice**

This value specifies the disk slice where the JumpStart program places the file system you want to duplicate. The slice value must follow the format cwtxdysz, for example c0t0d0s0 or c0t0d0s5. The JumpStart program creates a RAID-0 volume (single-slice concatenation) on the slice, and creates a RAID-1 volume to mirror the concatenation. You can specify up to two slices for two RAID-0 volumes.

**size**

This value specifies the size, in MB, of the file system.

**file-system**

This value specifies the file system that you are duplicating. The JumpStart program creates the RAID-1 volume from the slices that are specified and mounts the RAID-1 volume on the specified file system. In addition to critical file systems, such as root (/), /usr, and /var, you can also specify swap as the file system.

- 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-point**: Specifies the file system's mount-point name, for example, /var.
  - **swap**: Defines the slice to be used as swap.
  - **overlap**: Defines the slice 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. You can specify overlap only when **size** is one of the following values:
    - **existing**
    - **all**
    - **start:size**.
  - **unnamed**: Defines the slices as a raw slice. Therefore, the slice does not have a mount-point name. If you do not specify **file-system**, **unnamed** is used by default.
  - **ignore**: Specifies that the slice is not to be 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 the partitioning keyword and the existing value is specified.

**optional-parameters**  
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 file system. 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`.

**Note** – If unnamed is set for the `file-system` value, you cannot specify the `optional-parameters` value. See `file-system` for the values that can be set.

For more information about creating mirrored file systems during your installation, see Chapter 8, “Creating RAID-1 Volumes (Mirrors) During Installation (Overview),” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.

**forced_deployment Profile Keyword (Installing Flash Archive Differential Archives)**

`forced_deployment`

`forced_deployment` forces the installation of a Flash Archive 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 (UFS and ZFS)**

**Note** – The `geo` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations. For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.

`geo region`

go designates the regional locale or locales that you want to install on a system or to add when upgrading a system. `region` designates a geographical area that contains the locales that you want to install. Values you can specify for `region` 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>
<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>

For a complete list of the component locale values that compose each regional locale, see *International Language Environments Guide*.

**Note** – You can specify a geo keyword for each locale you need to add to a system.
**install_type Profile Keyword (UFS and ZFS)**

**Note** – The `install_type` keyword can be used for either a UFS file system or a ZFS root pool installation. The usage is limited for a ZFS installation. You can only use the `initial_install` option for a ZFS installation.

- If you want to migrate your UFS file system to a ZFS root pool or upgrade a ZFS root pool, you must use Live Upgrade. See Chapter 10, “Live Upgrade and ZFS (Overview),” in *Oracle Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning*.
- For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.
- For a description of how the `install_type` keyword can be used when installing a ZFS root pool, see “JumpStart Keywords for a ZFS Root (/) File System (Reference)” on page 151.

The `install_type` keyword defines whether to erase and install a new Oracle Solaris OS on a system, upgrade the existing Oracle Solaris OS on a system, or install a 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 `type`:

- `initial_install` specifies to perform an initial installation of the Oracle Solaris OS.
- `upgrade` specifies to perform an upgrade of the Oracle Solaris OS.
- `flash_install` specifies to install a flash archive that overwrites all files.
- `flash_update` specifies to install a Flash Archive differential archive that overwrites only the files that are specified.

**Note** – Each of these options has a specific set of related profile keywords.

**layout_constraint Profile Keyword**

`layout_constraint slice constraint minimum-size`

`layout_constraint` designates the constraint that auto-layout has on a file system if auto-layout needs to reallocate space during an upgrade because of space problems.
This keyword is used only with the upgrade option. You can use `layout_constraint` only for the upgrade option when you need to reallocate disk space.

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.

If the file system is not marked changeable, 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.

If file systems require more space for upgrade, 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.

<table>
<thead>
<tr>
<th>slice</th>
<th>Specifies the file system’s disk slice on which to specify the constraint. You must specify the system’s disk slice in the form <code>cwtxdysz</code> or <code>cxdysz</code>.</th>
</tr>
</thead>
</table>
| 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 MB, the changed size is 110 MB. 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.

**EXAMPLE 8–11**  
layout_constraint Profile Keyword

    layout_constraint c0t3d0s1 changeable 200
    layout_constraint c0t3d0s4 movable
    layout_constraint c0t3d1s3 available
    layout_constraint c0t2d0s1 collapse

**local_customization Profile Keyword (Installing Flash Archives)**

    local_customization local-directory
Before you install a 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 about predeployment and postdeployment scripts, see “Creating Customization Scripts” in Oracle Solaris 10/13 Installation Guide: Flash Archives (Creation and Installation).

### locale Profile Keyword (UFS and ZFS)

**Note** – The `locale` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations. For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.

`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. *International Language Environments Guide* 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.

### metadb Profile Keyword (Creating State Database Replicas)

`metadb slice [size size-in-blocks] [count number-of-replicas]`

The `metadb` keyword enables you to create Solaris Volume Manager state database replicas (mediates) during your JumpStart installation. You can use the `metadb` keyword multiple times in your profile file to create state database replicas on different disk slices.

`slice` You must specify the disk slice on which you want the JumpStart program to place the state database replica. The `slice` value must follow the format `cwtxdys`. 
The size optional keyword enables you to specify the size, in blocks, of the state database replica to be created. If you do not specify a size value, the JumpStart program uses a default size of 8192 blocks for the state database replica.

You can specify the number of state database replicas you are creating by setting the optional count keyword value in your profile. If you do not specify a count value, the JumpStart program creates three state database replicas by default.

For more information about creating Solaris Volume Manager state database replicas during your installation, see “State Database Replicas Guidelines and Requirements” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.

When installing a clone system with a Flash Archive differential archive, you can use the no_content_check keyword to ignore file-by-file validation. File-by-file validation ensures that the clone system is a duplicate of the master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.

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.

For information about installing Flash Archive differential archives, see “To Prepare to Install a Flash Archive With a JumpStart Installation” on page 77.

When installing a clone system with a Flash Archive 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 about installing Flash Archive differential archives, see “To Prepare to Install a Flash Archive With a JumpStart Installation” on page 77.

Profile Keywords and Values

| size size-in-blocks | The size optional keyword enables you to specify the size, in blocks, of the state database replica to be created. If you do not specify a size value, the JumpStart program uses a default size of 8192 blocks for the state database replica. |
| count number-of-replicas | You can specify the number of state database replicas you are creating by setting the optional count keyword value in your profile. If you do not specify a count value, the JumpStart program creates three state database replicas by default. |

no_content_check Profile Keyword (Installing Flash Archives)

no_check

When installing a clone system with a Flash Archive differential archive, you can use the no_content_check keyword to ignore file-by-file validation. File-by-file validation ensures that the clone system is a duplicate of the master system. Avoid using this keyword unless you are sure the clone system is a duplicate of the original master system.

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.

For information about installing Flash Archive differential archives, see “To Prepare to Install a Flash Archive With a JumpStart Installation” on page 77.

no_master_check Profile Keyword (Installing Flash Archives)

no_master_check

When installing a clone system with a Flash Archive 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 about installing Flash Archive differential archives, see “To Prepare to Install a Flash Archive With a JumpStart Installation” on page 77.

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.

### iSCSI Profile Keyword

When you are installing the Oracle Solaris OS on an iSCSI system, use the iSCSI keyword. See Example 3–16 and Example 3–17.

The following terms are defined for an iSCSI JumpStart installation.

**iSCSI Target IP**

This is required when configuring iSCSI over TCP/IP. Each target IP may represent a physical device, such as a disk or a tape or a virtualized target.

**iSCSI Target Port**

A port on which an iSCSI target server would listen on for requests from the Initiator. The default values is 3260.

**iSCSI Target LUN**

LUN (Logical Unit Number) represents an individually addressable (logical) SCSI device that is a part of a physical SCSI device (target). In an iSCSI environment, LUNs are essentially numbered disk drives. An initiator negotiates with a target to establish connectivity to a LUN. The result is an iSCSI connection that emulates a connection to a SCSI hard disk. Initiators treat iSCSI LUNs the same way as they would a raw SCSI or IDE hard drive For instance, rather than mounting remote directories as in NFS or CIFS environments, iSCSI systems format and directly manage filesystems on iSCSI LUNs. In enterprise deployments, LUNs usually represent slices of large RAID disk arrays, often allocated one per client. iSCSI imposes no rules or restrictions on multiple computers sharing individual LUNs. It leaves shared access to a single underlying filesystem as a task for the operating system.

LUN is required if the method if Static Discovery method is used for discovering the targets.

**iSCSI Target Name**

The iSCSI target name includes a iSCSI Qualified Name (IQN) which is documented in RFC 3720, with more examples of names in RFC 3721. See the IETF site for details. For example, `iqn.1986-03.com.sun:02:358ddbf8-6b1a-e73a-df56-89`, is a typical iSCSI target name. The name consists of the following fields:

- Literal IQN
- Date (yyyy-mm) that the naming authority took ownership of the domain
Reversed domain name of the authority
- Optional colon ":" prefixing a storage target name specified by the naming authority.

**package Profile Keyword (UFS and ZFS)**

*Note* – The package keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations. For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2.

```
package package-name [add [retrieval-type location]]| delete]
```

You can use package with both the initial installation and upgrade options. The package keyword enables you to do the following:
- Add a package to the software group from the Oracle Solaris distribution that is to be installed.
- Add a package to the software group from outside the distribution that is being installed.
- Exclude or remove a package from the software group that is to be installed or upgraded.
- Add a package from outside the distribution that is being installed when installing a flash archive.

**package-name**

Specifies the 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**

Specifies to add or remove the specified package. If you do not specify add or delete, add is used by default.

*Note* – You can add more than one package by adding another package entry to the profile and omitting the location. The location of the previous package is used for all subsequent packages if the location is left blank.

**[retrieval-type location]**

Specifies the addition of a package or packages that are located outside the Oracle Solaris distribution that is being installed. The values of retrieval-type and location depend on where the package is stored.

The following sections describe the values you can use for retrieval-type and location and examples of how to use the package_name keyword.
Packages Stored on an NFS Server

If the package is stored on an NFS server, use one of the following syntaxes for the package keyword.

```plaintext
package package-name add nfs server-name:/path [retry n]
pkg package add nfs://server-name:/path [retry n]
```

- **package-name**: Specifies the package name in the form SUNWname. To view detailed information about packages and their names, on an installed system, use the pkginfo -l command.
- **server-name**: Specifies the name of the server where you stored the package.
- **path**: Specifies the location of the package directory on the specified server. If the path contains $HOST, $HOST is replaced with the name of the host system that you are installing.
- **retry n**: Is an optional keyword. n is the maximum number of times the installation process attempts to mount the directory.

**EXAMPLE 8-12** Adding a Package by Using NFS

In this example, the package profile keyword adds the SUNWnew package from the NFS location nfs://golden/packages/Solaris_10/. If a mount fails, the NFS mount is tried five times.

```plaintext
package SUNWnew add nfs golden:/packages/Solaris_10 retry 5
```

Packages Stored on an HTTP Server

If the package is stored on an HTTP server, use one of the following syntaxes for the package keyword.

```plaintext
package package-name add http://server-name[:port] path optional-keywords
package package-name add http server-name[:port] path optional-keywords
```

- **package-name**: Specifies the package name in the form SUNWname. To view detailed information about packages and their names, on an installed system, use the pkginfo -l command.
- **server-name**: Specifies the name of the server where you stored the package.
- **port**: Specifies an optional port. port can be a port number or the name of a TCP service that has a port number that is determined at runtime.
  - If you do not specify a port, the default HTTP port number 80 is used.
- **path**: Specifies the location of the package to be retrieved from the specified server. When using an HTTP server, the package must be in package datastream format.
optional-keywords  Specifies the optional keywords to use when you retrieve a package from an HTTP server.

**TABLE 8–6**  Optional package Keywords to Use With HTTP

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout min</td>
<td></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. If a timeout occurs, the connection is closed, reopened, and resumed. If you specify a timeout value of 0 (zero), the connection is not reopened. If a timeout reconnection occurs, the package is retrieved from the beginning of the package and the data that was retrieved prior to the timeout is discarded.</td>
</tr>
<tr>
<td>proxy host:port</td>
<td></td>
<td>The proxy keyword enables you to specify a proxy host and proxy port. You can use a proxy host to retrieve an Oracle Solaris package from the other side of a firewall. You must supply a proxy port when you specify the proxy keyword.</td>
</tr>
</tbody>
</table>

**EXAMPLE 8–13**  Adding a Package by Using HTTP

In this example, the package profile keyword adds all the packages listed in the Solaris 10 directory from the HTTP location http://package.central/Solaris_10. If five minutes pass and no data is received, the package data is retrieved again. Previous package data is discarded. Either of the following formats can be used.

```
package SUNWnew add http://package.central/Solaris_10 timeout 5
```

**EXAMPLE 8–14**  Adding a Package by Using HTTP with a Proxy Port

In this example, the package profile keyword adds all the packages listed in the Solaris_10 directory from the HTTP location http://package.central/Solaris_10. The package is retrieved across a firewall by using the proxy keyword.

```
package SUNWnew add http://package.central/Solaris_10 proxy webcache.east:8080
```

**Packages Stored on a Local Device**

You can retrieve an Oracle Solaris package from a local device if you stored the package on a file system-oriented, random-access device, such as a diskette or a DVD-ROM. Use the following syntax for the package keyword:

```
package  package-name  add  local_device  device  path  file-system-type
```

`package-name`  Specifies the package name in the form SUNWname. To view detailed information about packages and their names, on an installed system, use the pkginfo -l command.
device Specifies the name of the drive where the Oracle Solaris package resides. 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 installation utility adds /dev/dsk/ to the path.

path Specifies the path to the Oracle Solaris package, relative to the root (/) file system on the device you specified.

file-system-type Specifies the type of file system on the device. If you do not supply a file system type, the installation utility attempts to mount a UFS file system. If the UFS mount fails, the installation utility attempts to mount an HSFS file system.

**EXAMPLE 8–15 Adding a Package by Using a Local Device With a UFS File System**

In this example, the package profile keyword adds the SUNWnew package from the directory /Solaris_10/Product from the local device c0t6d0s0. This is a UFS file system.

```bash
package SUNWnew add local_device c0t6d0s0 /Solaris_10/Product ufs
```

**EXAMPLE 8–16 Adding a Package by Using a Local Device From an HSFS File System**

In this example, the package profile keyword adds the SUNWnew package from the directory /Solaris_10/Product from the local device c0t6d0s0. This is an HSFS file system.

```bash
package SUNWnew add local_device c0t6d0s0 /Solaris_10/Product hsfs
```

**Packages Stored on a Local File**

A package can be installed from the miniroot from which you booted the system. When you perform a JumpStart installation, you boot the system from a DVD, CD, or an NFS-based miniroot. The installation software is loaded and run from this miniroot. Therefore, a package that you stored in the DVD, CD, or NFS-based miniroot is accessible as a local file. Use the following syntax for the package keyword.

```bash
package package-name add local_file path
```

**package-name** Specifies the package name in the form SUNWname. To view detailed information about packages and their names, on an installed system, use the pkginfo -l command.

**path** Specifies the location of the package. The path must be accessible to the system as a local file while the system is booted from the Oracle Solaris Software - 1 CD or from the Oracle Solaris Operating System DVD. The system cannot access /net when it is booted from the Oracle Solaris Software - 1 CD or from the Oracle Solaris Operating System DVD.
EXAMPLE 8–17  Adding a Package by Using a Local File

In this example, the package profile keyword adds the SUNWnew package from the /Solaris_10/Product directory.

```bash
package SUNWnew add local_file /Solaris_10/Product
```

Limitations When Using the package Keyword

Note these limitations when using the package keyword:

- 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.
- Packages cannot be retrieved from an FTP server location or local backup, such as tape.
- Packages within the Oracle Solaris distribution being installed cannot be added from alternate locations. If a package from the Oracle Solaris distribution is specified, the package cannot be followed by an alternative location in order to maintain consistency with the resulting installed system.
- In order to install without manual intervention, the package must be installable by using the pkgadd command. The same admin file must be used to install the software group packages and the package that resides in another location.
  - If the retrieval-type is HTTP, then the package must be in stream format.
  - If the retrieval-type is NFS server, local device, or local file, then the package should follow standard packaging format with the directory name being the same as the package being installed.
  - If a package is being added from a separate location and a package depends on another package that is not currently installed, the package is not installed. An error message is logged into the install or upgrade log file.
- If the package is being installed with a flash archive, follow these guidelines.
  - Any package installed must be compatible with the archive.
  - If a package is present in the archive, the JumpStart overwrites the existing package.

Upgrade Behavior When Using the package 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.

If you do not specify `partitioning` in the profile, the default type of partitioning is used by default.

`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` 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 Oracle 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.

**patch Profile Keyword**

*patch* `patch-ID-list` | `patch-file patch-location optional-keywords`

Chapter 8  •  JumpStart Keyword Reference 137
**patch-ID-list**

Specifies the patch ID numbers that are to be installed. The list should consist of comma-separated Oracle Solaris patch IDs. The patches are installed in the order specified in the list. Do not add a space after the comma, for example: 112467-01,112765-02.

**patch-file**

A file with a list of patches that is found in the *patch-location*. The patches are installed in the order specified in the file.

**patch-location**

Specifies the location where the patches reside. The following locations are allowed:

- NFS server
- HTTP server
- Local device
- Local file

**optional-keywords**

Optional keywords depend on where patches are stored. The following sections describe the possible locations and optional keywords.

### Patches Stored on an NFS Server

If the patch is stored on an NFS server, use one of the following options for the *patch* keyword.

```
patch patch-ID-list | patch-file nfs server-name/patch-directory [retry n]
patch patch-ID-list | patch-file nfs://server-name/patch-director [retry n]
```

**patch-ID-list**

Specifies the patch ID numbers that are to be installed. The list should be a list of comma-separated Oracle Solaris patch IDs. The patches are installed in the order specified in the list.

**patch-file**

A file with a list of patches that is found in the *patch-location*. The patches are installed in the order specified in the file.

**server-name**

Specifies the name of the server where you stored the patches.

**patch-directory**

Specifies the location of the patch directory on the specified server. The patches must be in standard patch format.

**retry n**

An optional keyword. *n* is the maximum number of times the install utility attempts to mount the directory.

**EXAMPLE 8–18  Adding a Patch With an Ordered List by Using NFS**

In this example, the *patch* profile keyword adds all the patches listed in the *patch* file from the NFS patch directory nfs:///patch_master/Solaris/v10/patches. Patches are installed in the order listed in the *patch*. If a mount fails, the NFS mount is tried five times.

```
patch patch_file nfs:///patch_master/Solaris/v10/patches retry 5
```
EXAMPLE 8–19 Adding a Patch by Using NFS

In this example, the patch profile keyword adds the patches 112467–01 and 112765–02 from the patch directory /Solaris/v10/patches on the server patch_master.

patch 112467-01,112765-02 nfs patch_master:/Solaris/v10/patches

Patches Stored on an HTTP Server

If the patch is stored on an HTTP server, use the following options for the patch keyword.

patch  patch-ID-list | patch-file http server-name [:port] patch-directory optional-http-keywords

patch-ID-list Specifies the patch ID numbers that are to be installed. The list should consist of comma-separated Oracle Solaris patch IDs. The patches are installed in the order specified in the list. Do not add a space after the comma, for example: 112467-01,112765-02.

patch-file A file with a list of patches that is found in the patch-location. The patches are installed in the order specified in the file.

server-name Specifies the name of the server where you stored the patch.

port Specifies an optional port. port can be a port number or the name of a TCP service that has a port number that is determined at runtime.

If you do not specify a port, the default HTTP port number 80 is used.

patch-directory Specifies the location of the patch directory to be retrieved from the specified server. When using an HTTP server, the patch must be in JAR format.

optional-keywords Specifies the optional keywords to use when you retrieve a patch from an HTTP server.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value Definition</th>
</tr>
</thead>
<tbody>
<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. If a timeout occurs, the connection is closed, reopened, and resumed. If you specify a timeout value of 0 (zero), the connection is not reopened. If a timeout reconnection occurs, the package is retried from the beginning of the package and the data that was retrieved prior to the timeout is discarded.</td>
</tr>
</tbody>
</table>
### TABLE 8–7 Optional patch Keywords to Use With HTTP (Continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Value Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy host:port</td>
<td>The proxy keyword enables you to specify a proxy host and proxy port. You can use a proxy host to retrieve an Oracle Solaris package from the other side of a firewall. You must supply a proxy port when you specify the proxy keyword.</td>
</tr>
</tbody>
</table>

**EXAMPLE 8–20** Adding a Patch With an Ordered List by Using HTTP

In this example, the patch profile keyword adds all the patches listed in the patch_file file from the HTTP location http://patch.central/Solaris/v10/patches. The patches are installed in the order specified in the file the patch file. If five minutes pass and no data is received, the patch data is retrieved again. Previous patch data is discarded.

```
patch patch_file http://patch.central/Solaris/v10/patches timeout 5
```

**EXAMPLE 8–21** Adding a Patch by Using HTTP

In this example, the patch profile keyword entry adds the patches 112467–01 and 112765–02 from the patch location http://patch_master/Solaris/v10/patches.

```
patch 112467-01,112765-02 http://patch.central/Solaris/v10/patches
```

### Patches Stored on a Local Device

You can retrieve an Oracle Solaris package from a local device if you stored the package on a file system-oriented, random-access device, such as a diskette or a DVD-ROM. Use the following syntax for the patch keyword.

```
patch patch-ID-list | patch-file local_device \
    device path file-system-type
```

- **patch-ID-list**: Specifies the patch ID numbers that are to be installed. The list should consist of comma-separated Oracle Solaris patch IDs. The patches are installed in the order specified in the list. Do not add a space after the comma, for example: 112467-01,112765-02.
- **patch-file**: A file with a list of patches that is found in patch-location. The patches are installed in the order specified in the file.
- **device**: Specifies the name of the drive where the Oracle Solaris package resides. 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 installation utility adds /dev/dsk/ to the path.
- **path**: Specifies the path to the Oracle Solaris patch, relative to the root (/) file system on the device you specified.
file-system-type

Specifies the type of file system on the device. If you do not supply a file system type, the installation utility attempts to mount a UFS file system. If the UFS mount fails, the installation utility attempts to mount an HSFS file system.

EXAMPLE 8-22  Adding a Patch With an Ordered List by Using a Local Device

In this example, the patch profile keyword adds all the patches listed in the patch_file file from the directory /Solaris_10/patches from the local device c0t6d0s0. The patch file determines the order of patches to be installed.

patch patch_file c0t6d0s0 /Solaris_10/patches

EXAMPLE 8-23  Adding a Patch by Using a Local Device

In this example, the patch profile keyword adds the patches 112467–01 and 112765–02 from the patch directory /Solaris_10/patches from local device c0t6d0s0.

patch 112467-01,112765-02 local_device c0t6d0s0 /Solaris_10/patches

Patches Stored on a Local File

A patch can be installed from the miniroot from which you booted the system. When you perform a JumpStart installation, you boot the system from a DVD, CD, or NFS-based miniroot. The installation software is loaded and run from this miniroot. Therefore, a patch that you stored in the DVD, CD, or NFS-based miniroot is accessible as a local file. Use the following syntax for the patch keyword.

patch  patch-ID-list  |  patch-file  local_file  patch-directory

patch-ID-list

Specifies the patch ID numbers that are to be installed. The list should consist of comma-separated Oracle Solaris patch IDs. The patches are installed in the order specified in the list. Do not add a space after the comma, for example: 112467-01,112765-02.

patch-file

A file with a list of patches that is found in the patch-location. The patches are installed in the order specified in the file.

patch-directory

Specifies the location of the patch directory. The patch directory must be accessible to the system as a local file while the system is booted from the Oracle Solaris Software - 1 CD or from the Oracle Solaris Operating System DVD. The system cannot access /net when it is booted from the Oracle Solaris Software - 1 CD or from the Oracle Solaris Operating System DVD.
In this example, the `patch` profile keyword adds all the patches that are listed in the `patch_file` file from the `/Solaris_10/patches` directory. The `patch_file` determines the order of patches to be installed.

```
patch patch_cal_file local_file /Solaris_10/patches
```

In this example, the `patch` profile keyword adds the patches 112467–01 and 112765–02 from the patch directory `/Solaris_10/patches`.

```
patch 112467-01,112765-02 local_file /Solaris_10/patches
```

**Limitations When Using the `patch` Keyword**

Note the following limitations when using the `patch` keyword:

- Patches cannot be retrieved from FTP locations or local backup, such as tape.
- Signed patches cannot be added.
- Patches must be installable with the `patchadd` command.
- If a patch depends on a patch that is not currently installed, the patch is not installed. An error message is logged into the installation or upgrade log file.
- You must determine the correct order of the patches for a correct installation of the patches.

**pool Profile Keyword (ZFS Only)**

The `pool` keyword defines the installation of a ZFS root pool. The pool is installed with a software group specified with the `cluster` keyword. The `poolsize`, `swapsize`, `dumpsize`, and `vdevlist` options are needed for creating a new root pool.

For a complete description of the `pool` keyword and other keywords that can be used for a ZFS root pool, see "pool Profile Keyword (ZFS Only)" on page 153.

**root_device Profile Keyword (UFS and ZFS)**

**Note** – The `root_device` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the limited to a single system for ZFS installations.

- For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2
- For a description of how the `root_device` keyword can be used when installing a ZFS root pool, see "JumpStart Keywords for a ZFS Root (/) File System (Reference)" on page 151
root_device slice


Note—The root disk is determined by the JumpStart program and determines where the OS is to be installed. The rules file uses a probe keyword “rootdisk,” but this keyword is used differently than the rootdisk keyword used in the JumpStart profile. You cannot set the place of installation by using the probe keyword “rootdisk” in the rules file. The probe keyword, rootdisk, determines where to boot from during the installation. See Table 8–10.

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 cwtxysz or cxdysz.

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 RAID-1 volume (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.

EXAMPLE 8–26 root_device Profile Keyword
root_device c@0d0s2

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. The following table describes how the JumpStart program determines the system's root disk for the installation.

Note — The JumpStart program determines a system's root disk size only during an initial installation. You cannot change a system’s root disk during an upgrade.
### TABLE 8–8 How JumpStart Determines a System’s Root Disk (Initial Installation)

<table>
<thead>
<tr>
<th>Stage</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If the <code>root_device</code> keyword is specified in the profile, the JumpStart program sets <code>rootdisk</code> to the root device.</td>
</tr>
<tr>
<td>2</td>
<td>If <code>rootdisk</code> is not set and the <code>boot_device</code> keyword is specified in the profile, the JumpStart program sets <code>rootdisk</code> to the boot device.</td>
</tr>
<tr>
<td>3</td>
<td>If <code>rootdisk</code> is not set and a <code>filesys cwtxdysz size / entry</code> is specified in the profile, the JumpStart program sets <code>rootdisk</code> to the disk that is specified in the entry.</td>
</tr>
<tr>
<td>4</td>
<td>If <code>rootdisk</code> is not set and a <code>rootdisk.sn</code> entry is specified in the profile, the JumpStart program searches the system's disks in kernel probe order for an existing root file system on the specified slice. If a disk is found, the JumpStart program sets <code>rootdisk</code> to the found disk.</td>
</tr>
<tr>
<td>5</td>
<td>If <code>rootdisk</code> is not set and <code>partitioning existing</code> is specified in the profile, the JumpStart program searches the system's disks in kernel probe order for an existing root file system. If a root file system is not found or more than one is found, an error occurs. If a root file system is found, the JumpStart program sets <code>rootdisk</code> to the found disk.</td>
</tr>
<tr>
<td>6</td>
<td>If <code>rootdisk</code> is not set, the JumpStart program sets <code>rootdisk</code> 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 Oracle Solaris OS is to be installed.

`type-switch` represents the option `standalone` or `server`, which you use to indicate the type of system on which the Oracle Solaris software is to be installed. If you do not specify `system_type` in a profile, `standalone` is used by default.

### usedisk Profile Keyword (UFS and ZFS)

The `usedisk` keyword can be used when you install either a UFS file system or a ZFS root pool. The usage for this keyword is the same in both UFS and ZFS installations.

- For a complete list of keywords that can be used in a UFS or ZFS installation, see Table 8–2
- For information on performing a ZFS installation, see Chapter 9, “Installing a ZFS Root Pool With JumpStart”

**usedisk** `disk-name` ...

Each disk instance for this keyword requires a separate line item. Specify multiple disks for usage on separate lines, as shown in the following example:

```
usedisk c0t0d0
usedisk c0t1d0
usedisk c0t2d0
```
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 cxytzd 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 don't use keyword in the same profile.

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_DISKSIZE, 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.

The following table describes these variables and their values.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<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_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 &quot;How the System's Root Disk Is Determined&quot; on page 143.</td>
</tr>
<tr>
<td>SI_DISKSIZE</td>
<td>A comma-separated list of disk sizes on the install client. The SI_DISKSIZE variable is set when the disksize keyword is used and matched in the rules file.</td>
</tr>
<tr>
<td>SI_DOMAINNAME</td>
<td>The domain name. The SI_DOMAINNAME variable is set when the domainname 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>
</tbody>
</table>
### Table 8–9: Installation Environment Variables (Continued)

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI_HOSTADDRESS</td>
<td>The install client's IP 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>
<tr>
<td>SI_INSTALLED</td>
<td>The device name of a disk with a specific operating system on the disk, for example, Oracle Solaris 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 &quot;How the System's Root Disk Is Determined&quot; on page 143.</td>
</tr>
<tr>
<td>SI_OSNAME</td>
<td>The operating system release on the Oracle Solaris software image. For example, you can use the SI_OSNAME variable in a script if you are installing the Oracle Solaris software on systems that are based on the version of the operating system on the Oracle Solaris Operating System DVD or the Oracle Solaris Software - 1 CD image.</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. The SI_ROOTDISK variable sets the device to boot from during the installation.</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>
</tbody>
</table>

Note – You cannot set the place of installation by using the probe keyword rootdisk in the rules file. For information on the rootdisk variable that is set in a JumpStart profile, see "How the System's Root Disk Is Determined" on page 143.
### Probe Keywords and Values

The following table describes each rule keyword and its equivalent probe keyword.

**Note** – Always place probe keywords at or near the beginning of the rules file.

#### TABLE 8–10 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>Determines the kernel architecture, i386 or SPARC, and sets SI_ARCH.</td>
</tr>
<tr>
<td>arch</td>
<td>arch</td>
<td>Returns the size of a system’s disks in MB in kernel probe order, c0t3d0s0, c0t3d0s1, c0t4d0s0. disksizes SI_DISKLIST, SI_DISKSIZES, SI_NUMDISKS, and SI_TOTALDISK.</td>
</tr>
<tr>
<td>disksize</td>
<td>disks</td>
<td>Returns a system’s NIS or NIS+ domain name or blank and sets SI_DOMAINNAME. The domainname keyword returns the output of <code>domainname(1M)</code>.</td>
</tr>
<tr>
<td>domainname</td>
<td>domainname</td>
<td>Returns a system’s host name that is the output from <code>uname(-n)</code> and sets SI_HOSTNAME.</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 <code>ifconfig(1M)</code> -a that is not lo0, and sets SI_HOSTADDRESS.</td>
</tr>
<tr>
<td>hostname</td>
<td>hostname</td>
<td>Returns a system’s platform name and sets SI_MODEL. For a list of platform names, see the Oracle Solaris Sun Hardware Platform Guide at <a href="http://www.oracle.com/technetwork/indexes/documentation/index.html">http://www.oracle.com/technetwork/indexes/documentation/index.html</a>.</td>
</tr>
<tr>
<td>installed</td>
<td>installed</td>
<td>Returns the version name of the Oracle Solaris OS that is installed on a system and sets SI_ROOTDISK and SI_INSTALLED. If the JumpStart program finds an Oracle Solaris release but is unable to determine the version, the version that is returned is SystemV.</td>
</tr>
<tr>
<td>karch</td>
<td>karch</td>
<td>Returns a system’s platform group, for example 186pc or sun4u, and sets SI_KARCH. For a list of platform names, see the Oracle Solaris Sun Hardware Platform Guide at <a href="http://www.oracle.com/technetwork/indexes/documentation/index.html">http://www.oracle.com/technetwork/indexes/documentation/index.html</a>.</td>
</tr>
<tr>
<td>memsize</td>
<td>memsize</td>
<td>Returns the size of physical memory on a system in MB 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 Oracle Solaris Sun Hardware Platform Guide at <a href="http://www.oracle.com/technetwork/indexes/documentation/index.html">http://www.oracle.com/technetwork/indexes/documentation/index.html</a>.</td>
</tr>
</tbody>
</table>
### TABLE 8-10 Descriptions of Probe Keywords  
*Continued*

<table>
<thead>
<tr>
<th>Rule Keyword</th>
<th>Equivalent Probe Keyword</th>
<th>Description of Probe Keyword</th>
</tr>
</thead>
<tbody>
<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 <code>ifconfig(1M)</code> - a that is not <code>lo0</code>. The <code>network</code> keyword sets <code>SI_NETWORK</code>.</td>
</tr>
<tr>
<td>osname</td>
<td>osname</td>
<td>Returns the version and operating system name of the Oracle Solaris OS that is found on a CD and sets <code>SI_OSNAME</code>. If the JumpStart program finds an Oracle Solaris release but is unable to determine the version, the version that is returned is <code>SystemV</code>.</td>
</tr>
<tr>
<td>rootdisk</td>
<td></td>
<td>Returns the name and size in MB of a system's root disk and sets <code>SI_ROOTDISK</code>.</td>
</tr>
<tr>
<td>totaldisk</td>
<td>totaldisk</td>
<td>Returns the total disk space on a system (in MB) and sets <code>SI_TOTALDISK</code>. The total disk space includes all of the operational disks that are attached to a system.</td>
</tr>
</tbody>
</table>
Installing a ZFS Root Pool With JumpStart

This chapter provides the information necessary for performing a JumpStart installation for a ZFS root pool. The following sections provide planning information, profile examples, and profile keyword descriptions:

- “JumpStart Installation for a ZFS Root (/) File System (Overview and Planning)” on page 150
- “JumpStart Profile Examples for a ZFS Root Pool” on page 154
- “JumpStart Keywords for a ZFS Root (/) File System (Reference)” on page 151

What's New in the Solaris 10 10/09 Release

Starting with the Solaris 10 10/09 release, you can set up a JumpStart profile to identify a flash archive of a ZFS root pool.

A flash archive can be created on a system that is running a UFS root file system or a ZFS root file system. A flash archive of a ZFS root pool contains the entire pool hierarchy except for the swap and dump volumes and any excluded datasets. The swap and dump volumes are created when the flash archive is installed.

You can use the Flash Archive installation method as follows:

- Generate a flash archive that can be used to install and boot a system with a ZFS root file system.
- Perform a JumpStart installation of a system by using a ZFS flash archive.

Note – Creating a ZFS flash archive backs up an entire root pool, not individual boot environments. Individual datasets within the pool can be excluded by using the flar create and flar command’s -D option.

For detailed instructions and limitations, see “Installing a ZFS Root File System (Oracle Solaris Flash Archive Installation)” in Oracle Solaris ZFS Administration Guide.
JumpStart Installation for a ZFS Root (/) File System (Overview and Planning)

This section describes how to create a JumpStart profile to install a ZFS root pool.

Note – If you want to install a UFS root (/) file system, all existing profile keywords work as in previous Oracle Solaris releases. For a list of UFS profile keywords, see Chapter 8, “JumpStart Keyword Reference.”

A ZFS specific profile must contain the pool keyword. The pool keyword installs a new root pool and a new boot environment is created by default. You can create a separate /var dataset with existing bootenv installbe keywords and the new bename and dataset options. Some keywords that are allowed in a UFS-specific profile are not allowed in a ZFS specific profile, such as those specifying the creation of UFS mount points.


Limitations for a JumpStart Installation for a ZFS Root Pool

Keep the following issues in mind before considering a JumpStart installation of a bootable ZFS root pool.

<table>
<thead>
<tr>
<th>Limitation</th>
<th>Description</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>For a JumpStart installation, you cannot use an existing ZFS storage pool to create a bootable ZFS root pool.</td>
<td>You must create a new ZFS storage pool with syntax similar to the following: <code>pool rpool 20G 4G 4G c0t0d0s0</code></td>
<td>“pool Profile Keyword (ZFS Only)” on page 153</td>
</tr>
</tbody>
</table>
|             | The complete pool keyword line is required because you cannot use an existing pool. The bootenv keyword line is optional. If you do not use bootenv, a default boot environment is created for you. For example: `install_type initial_install
cluster SUNWCall
pool rpool 20G 4G any
bootenv installbe bename newBE` | |

Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations • January 2013
JumpStart Keywords for a ZFS Root (/) File System (Reference)

This section provides descriptions of some of the ZFS specific keywords that you can use in a JumpStart profile. The keywords described in this section are either used differently from their usage in a UFS profile or used only in a ZFS profile.

- For a quick reference guide to UFS and ZFS profile keywords, see "Profile Keywords Quick Reference" on page 102.
- The following keywords can be used in a ZFS profile. The usage is the same for both UFS and ZFS profiles. For descriptions of these keywords, see "Profile Keyword Descriptions and Examples" on page 103.
  - boot_device
  - cluster
  - dontuse
  - fdisk
  - filesys (mounting remote file systems)
  - geo
  - locale
  - package
  - usedisk

TABLE 9–1 JumpStart Limitations for ZFS Root Pools (Continued)

<table>
<thead>
<tr>
<th>Limitation Description</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>You cannot create a pool with whole disks.</td>
<td>You must create your pool with disk slices rather than whole disks.</td>
</tr>
<tr>
<td>If in the profile you create a pool with whole disks, such as c0t0d0, the installation fails. You will receive an error message similar to the following.</td>
<td>Invalid disk name (c0t0d0)</td>
</tr>
<tr>
<td>Some keywords that are allowed in a UFS specific profile are not allowed in a ZFS specific profile, such as those specifying the creation of UFS mount points.</td>
<td>&quot;Profile Keywords Quick Reference&quot; on page 102</td>
</tr>
<tr>
<td>You cannot upgrade with JumpStart. You must use Live Upgrade.</td>
<td>With Live Upgrade, you can create a copy of the currently running system. This copy can be upgraded and then activated to become the currently running system.</td>
</tr>
</tbody>
</table>
**bootenv Profile Keyword (ZFS and UFS)**

The `bootenv` keyword identifies boot environment characteristics. A boot environment is created by default during installation with the `pool` keyword. If you use the `bootenv` keyword with the `installbe` option, you can name the new boot environment and create a `/var` dataset within the boot environment.

This keyword can be used in a profile for installing a UFS file system or a ZFS root pool.

- In a UFS file system, this keyword is used for creating an empty boot environment for the future installation of a flash archive. For more information, see "bootenv Profile Keyword (UFS and ZFS)" on page 112.
- For a ZFS root pool, the `bootenv` keyword changes the characteristics of the default boot environment that is created at install time. This boot environment is a copy of the root file system that you are installing.

The `bootenv` keyword can be used with the `installbe`, `bename`, and `dataset` options. These options name the boot environment and create a separate `/var` dataset.

```
bootenv installbe bename BE-name [dataset mount-point]
```

- `installbe`: Changes the characteristics of the default boot environment that is created during the installation.
- `bename BE-name`: Specifies the name of the new boot environment to be created. The name can be no longer than 30 characters, can contain only alphanumeric characters, and can contain no multibyte characters. The name must be unique on the system.
- `dataset mount-point`: Use the optional dataset keyword to identify a `/var` dataset that is separate from the ROOT dataset. The `mount-point` value is limited to `/var`. For example, a bootenv syntax line for separate `/var` dataset would be similar to the following:

```
bootenv installbe bename zfsroot dataset /var
```

For more information about upgrading and activating a boot environment, see Chapter 10, “Live Upgrade and ZFS (Overview),” in *Oracle Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning.*

**install_type Keyword (ZFS and UFS)**

The `install_type` keyword is required in every profile. For a UFS installation, several options are available. The only option available for a ZFS installation is the `initial_install` keyword. This option installs a new Oracle Solaris OS on a system. The profile syntax is as follows:

```
install_type initial_install
```
The following UFS options are not available for a ZFS installation:

- **flash_install** - A flash archive cannot be installed.
- **flash_update** - A flash archive cannot be installed.

### pool Profile Keyword (ZFS Only)

The `pool` keyword defines the new root pool to be created. The pool is then installed with a software group specified with the `cluster` keyword. All of the options shown in the syntax are needed for creating a new root pool.

```
pool poolname poolsize swapsize dumpsize vdevlist
```

**Note** – If you provide a specific size value, it is assumed to be in MB unless specified by g (GB).

- **poolname**: Specifies the name of the new pool to be created. A new pool is created with the specified `poolsize` and with the specified devices, `vdevlist`.
- **poolsize**: Size of the new pool to be created. You can either specify a size or use the `auto` option to allocate the largest possible pool size given the constraints, such as size of the disks and preserved slices.

  **Note** – The meaning of `auto` for the `poolsize` keyword is different from the `filesys` keyword use of `auto` in a UFS file system. In ZFS, the size of the disk is checked to verify that the minimum size can be accommodated. If the minimum size is available, the largest possible pool size is allocated given the constraints, such as size of the disks and preserved slices.

- **swapsize**: Size of the swap volume (zvol) to be created within a new root pool. You can either provide a size or use the `auto` option to cause the swap area to be automatically sized. The default size is half the size of physical memory, but no less than 512 MB and no greater than 2 GB.
- **dumpsize**: Size of the dump volume to be created within a new pool. You can either provide a size or use the `auto` option to specify the default dump size.
- **vdevlist**: One or more devices used to create the pool.
Devices in the vdevlist must be slices for the root pool. Slice names should be provided in the form cwxtdyz.

**Note** – The format of the vdevlist is the same as the format of the zpool create command.

Acceptable values for this option are:

- A single device name For example, c0t0d0s0
- mirror [device-names|any] mirror specifies the mirroring of the disk with either specified device names or the keyword any, which enables the installer to select a suitable device.

At this time, only mirrored configurations are supported when multiple devices are specified. You can mirror as many as disks you like, but the size of the pool created is determined by the smallest of the specified disks. For more information about creating mirrored storage pools, see “Mirrored Storage Pool Configuration” in Oracle Solaris ZFS Administration Guide.

- any Enables the installer to select a suitable device.

**root_device Profile Keyword (ZFS and UFS)**

`root_device cwxtdyz`

`root_device` specifies the device to be used for the root pool. The `root_device` keyword determines where the operating system is installed. This keyword is used the same in both ZFS and a UFS file system with some limitations. For the ZFS root pool, the root device is limited to a single system. This keyword is not useful for mirrored pools.

**JumpStart Profile Examples for a ZFS Root Pool**

This section provides examples of ZFS specific JumpStart profiles.
Note – For the ZFS root pool to be upgradeable and bootable, you must create your pool with disk slices rather than whole disks. If in the profile you create a pool with whole disks, such as c0t0d0, you will receive an error message similar to the following.

Invalid disk name (c0t0d0)

EXAMPLE 9-1 Installing a Mirrored ZFS Root Pool

install_type initial_install
cluster SUNWCall
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename solaris10_6

The following keywords and values are used in this example.

install_type initial_install The install_type keyword is required in every profile. The initial_install keyword performs an initial installation that installs a new Oracle Solaris OS in a new ZFS root pool.

cluster The Entire Distribution software group, SUNWCall, is installed on the system. For more information about software groups, see “Disk Space Recommendations for Software Groups” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.

pool The pool keyword defines the characteristics of the new ZFS root pool.

newpool Defines the name of the root pool.

auto Specifies the size of the disks automatically. The size is determined by the size of the specified disks.

auto The swap area is automatically sized with the auto keyword. The default size is half the size of physical memory, but no less than 512 MB and no greater than 2 GB. You can set the size outside this range by using the size option.

auto The dump device is automatically sized.

mirror The mirrored configuration of disks has the mirror keyword and disk slices specified as c0t0d0s0 and c0t1d0s0.
EXAMPLE 9–1 Installing a Mirrored ZFS Root Pool  
(Continued)

bootenv

install be changes the characteristics of the default boot environment that is created during the installation.

rename

Names the new boot environment solaris10_6.

EXAMPLE 9–2 Customizing the Disk Size For a ZFS Root Pool

install_type initial_install

cluster SUNWCall

pool newpool 80g 2g 2g mirror any any

bootenv install be rename solaris10_6

The following keywords and values are used in this example.

install_type initial_install

The install_type keyword is required in every profile. The initial_install keyword performs an initial installation that installs a new Oracle Solaris OS in a new ZFS root pool.

cluster

The Entire Distribution software group, SUNWCall, is installed on the system. For more information about software groups, see “Disk Space Recommendations for Software Groups” in Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade.

pool

The pool keyword defines the characteristics of the new ZFS root pool.

newpool

Specifies the name of the root pool.

80g

Specifies the size of the disk slice.

2g

The swap area and dump volumes are 2 GB.

mirror

The mirrored configuration of disks has the mirror keyword and disk slices specified as c0t0d0s0 and c0t1d0s0.

The any options in the mirrored configuration finds any two available devices that are large enough to create a 80 GB pool. If two such devices are not available, the install fails.

bootenv

install be changes the characteristics of the default boot environment that is created during the installation.
EXAMPLE 9–2 Customizing the Disk Size For a ZFS Root Pool (Continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bename</code></td>
<td>Names the new boot environment <code>solaris10_6</code>.</td>
</tr>
</tbody>
</table>

EXAMPLE 9–3 Specifying Where to Install the OS

install_type initial_install
cluster SUNWCall
root_device c0t0d0s0
pool nrpool auto auto auto rootdisk.s0
bootenv installbe bename bnv dataset /var

The following keywords and values are used in this example.

install_type initial_install  The `install_type` keyword is required in every profile. The `initial_install` keyword performs an initial installation that installs a new Oracle Solaris OS in a new ZFS root pool.

cluster  The Entire Distribution software group, SUNWCall, is installed on the system. For more information about software groups, see “Disk Space Recommendations for Software Groups” in Oracle Solaris 10/13 Installation Guide: Planning for Installation and Upgrade.

root_device  Specifies the disk slice where the OS is to be installed. The `c0t0d0s0` defines the specific disk and slice for the OS.

pool  The `pool` keyword defines the characteristics of the new ZFS root pool.

nrpool  Defines the name of the root pool.

auto  Specifies the size of the disks automatically. The size is determined by the size of the specified disks.

auto  The swap area is automatically sized with the `auto` keyword. The default size is half the size of physical memory, but no less than 512 MB and no greater than 2 GB. You can set the size outside this range by using the `size` option.

auto  The dump device is automatically sized.

rootdisk.s0  The device used to create the root pool is specified as slice 0.
EXAMPLE 9–3  Specifying Where to Install the OS  (Continued)

bootenv

installbe changes the characteristics of the default boot environment that is created during the installation.

bename  Names the new boot environment bnv.

dataset  Creates a /var dataset that is separate from the ROOT dataset. /var is the only value for dataset.

Additional Resources

For additional information about the topics included in this chapter, see the following resources:

- For ZFS information, including overview, planning, and step-by-step instructions, see Oracle Solaris ZFS Administration Guide.
- For a list of all JumpStart keywords, see Chapter 8, “JumpStart Keyword Reference.”
- For information about using Live Upgrade to migrate from UFS to ZFS or create a new boot environment in a ZFS root pool, see Chapter 10, “Live Upgrade and ZFS (Overview),” in Oracle Solaris 10 1/13 Installation Guide: Live Upgrade and Upgrade Planning.
Glossary

archive  A file that contains a collection of files that were copied from a master system. The file also contains identification information about the archive, such as a name and the date that you created the archive. After you install an archive on a system, the system contains the exact configuration of the master system.

An archive could be a differential archive, which is a 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.

begin script  A user-defined Bourne shell script, specified within the rules file, that performs tasks before the Oracle Solaris software is installed on the system. You can use begin scripts only with JumpStart, a feature of Oracle Solaris, installations.

boot  To load the system software into memory and start it.

boot environment  A collection of mandatory file systems (disk slices and mount points) that are critical to the operation of the Oracle Solaris OS. These disk slices might be on the same disk or distributed across multiple disks.

The active boot environment is the one that is currently booted. Exactly one active boot environment can be booted. An inactive boot environment is not currently booted, but can be in a state of waiting for activation on the next reboot.

boot loader  x86 only: The boot loader is the first software program that runs after you turn on a system. This program begins the booting process.

boot server  A server system that provides client systems on the same network subnet with the programs and information that they need to start. A boot server is required to install over the network if the install server is on a different subnet than the systems on which Oracle Solaris software is to be installed.

Core Software Group  A software group that contains the minimum software that is required to boot and run the Oracle Solaris OS on a system. Core includes some networking software and the drivers that are required to run the Common Desktop Environment (CDE) desktop. Core does not include the CDE software.

custom probes file  A file, which must be located in the same JumpStart directory as the rules file, that is a Bourne shell script that contains two types of functions: probe and comparison. Probe functions gather the information you want or do the actual work and set a corresponding SI_environment variable you define. Probe functions become probe keywords. Comparison functions call a corresponding probe function, compare the output of the probe function, and return 0 if the keyword matches or 1 if the keyword doesn’t match. Comparison functions become rule keywords. See also rules file.
### dataset
A generic name for the following ZFS entities: clones, file systems, snapshots, or volumes.

### Developer Oracle Solaris Software Group
A software group that contains the End User Oracle Solaris Software Group plus the libraries, include files, man pages, and programming tools for developing software.

### DHCP
(Dynamic Host Configuration Protocol) An application-layer protocol. Enables individual computers, or clients, on a TCP/IP network to extract an IP address and other network configuration information from a designated and centrally maintained DHCP server or servers. This facility reduces the overhead of maintaining and administering a large IP network.

### differential archive
A latch archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files that are specified and is restricted to systems that contain software consistent with the unchanged master image.

### End User Oracle Solaris Software Group
A software group that contains the Core Software Group plus the recommended software for an end user, including the Common Desktop Environment (CDE) and DeskSet software.

### Entire Oracle Solaris Software Group
A software group that contains the entire Oracle Solaris release.

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

### /etc directory
A directory that contains critical system configuration files and maintenance commands.

### /export file system
A file system on an OS server that is shared with other systems on a network. For example, the /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.

### fdisk partition
A logical partition of a disk drive that is dedicated to a particular operating system on x86 based systems. To install the Oracle Solaris software, you must set up at least one Oracle Solaris 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 Oracle Solaris fdisk partition per disk.

### file server
A server that provides the software and file storage for systems on a network.

### file system
In the operating system, a tree-structured network of files and directories that you can access.

### finish script
A user-defined Bourne shell script, specified within the rules file, that performs tasks after the Oracle Solaris software is installed on the system but before the system reboots. You use finish scripts with JumpStart installations.
Flash Archive  A Oracle Solaris installation feature that enables you to create an archive of the files on a system, called 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.

format  To put data into a structure or divide a disk into sectors for receiving data.

function key  One of the 10 or more keyboard keys that are labeled F1, F2, F3, and so on that are mapped to particular tasks.

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

GRUB  x86 only: GNU GRand Unified Bootloader (GRUB) is an open source boot loader with a simple menu interface. The menu displays a list of operating systems that are installed on a system. GRUB enables you to easily boot these various operating systems, such as the Oracle Solaris OS, Linux, or Microsoft Windows.

GRUB edit menu  x86 only: A boot menu that is a submenu of the GRUB main menu. GRUB commands are displayed on this menu. These commands can be edited to change boot behavior.

GRUB main menu  x86 only: A boot menu that lists the operating systems that are installed on a system. From this menu, you can easily boot an operating system without modifying the BIOS or fdisk partition settings.

initial installation  An installation that overwrites the currently running software or initializes a blank disk.

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

install server  A server that provides the Oracle Solaris DVD or CD images from which other systems on a network can install Oracle Solaris (also called a media server). You can create an install server by copying the Oracle Solaris DVD or CD images to the server’s hard disk.

JumpStart directory  When you use a profile diskette for JumpStart installations, the JumpStart directory is the root directory on the diskette that contains all the essential JumpStart files. When you use a profile server for JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential JumpStart files.

JumpStart installation  A type of installation in which the Oracle Solaris software is automatically installed on a system by using the factory-installed JumpStart software.

Live Upgrade  An upgrade method that enables a duplicate boot environment to be upgraded while the active boot environment is still running, thus eliminating downtime of the production environment.

locale  A geographic or political region or community that shares the same language, customs, or cultural conventions (English for the U.S. is en_US, and English for the U.K. is en_UK).
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>mirror</td>
<td>See <strong>RAID-1 volume</strong>.</td>
</tr>
<tr>
<td>mount</td>
<td>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).</td>
</tr>
<tr>
<td>mount point</td>
<td>A workstation directory to which you mount a file system that exists on a remote machine.</td>
</tr>
<tr>
<td>networked systems</td>
<td>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.</td>
</tr>
<tr>
<td>non-global zone</td>
<td>A virtualized operating system environment created within a single instance of the Oracle Solaris operating system. One or more applications can run in a non-global zone without interacting with the rest of the system. Non-global zones are also called zones. See also Oracle Solaris Zones and global zone.</td>
</tr>
<tr>
<td>/opt file system</td>
<td>A file system that contains the mount points for third-party and unbundled software.</td>
</tr>
<tr>
<td>Oracle Solaris DVD or CD images</td>
<td>The Oracle Solaris software that is installed on a system, which you can access on the Oracle Solaris DVDs or CDs or an install server’s hard disk to which you have copied the Oracle Solaris DVD or CD images.</td>
</tr>
<tr>
<td>package</td>
<td>A collection of software that is grouped into a single entity for modular installation. The Oracle Solaris software is divided into software groups, which are each composed of clusters and packages.</td>
</tr>
<tr>
<td>platform group</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>platform name</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>pool</td>
<td>A logical group of devices describing the layout and physical characteristics of the available ZFS storage. Space for datasets is allocated from a pool.</td>
</tr>
<tr>
<td>probe keyword</td>
<td>A syntactical element that extracts attribute information about a system when using the 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>profile</td>
<td>A text file that defines how to install the Oracle Solaris software when using the JumpStart method. For example, a profile defines which software group to install. Every rule specifies a profile that defines how a system is to be installed when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule. See also rules file.</td>
</tr>
<tr>
<td>profile diskette</td>
<td>A diskette that contains all the essential JumpStart files in its root directory (JumpStart directory).</td>
</tr>
<tr>
<td>RAID-0 volume</td>
<td>A class of volume that can be a stripe or a concatenation. These components are also called submirrors. A stripe or concatenation is the basic building block for mirrors.</td>
</tr>
<tr>
<td>RAID-1 volume</td>
<td>A class of volume that replicates data by maintaining multiple copies. A RAID-1 volume is composed of one or more RAID-0 volumes called submirrors. A RAID-1 volume is sometimes called a mirror.</td>
</tr>
<tr>
<td><strong>Reduced Network Support Software Group</strong></td>
<td>A software group that contains the minimum code that is required to boot and run an Oracle Solaris system with limited network service support. The Reduced Networking Software Group provides a multiuser text-based console and system administration utilities. This software group also enables the system to recognize network interfaces, but does not activate network services.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>root</strong></td>
<td>The top level of a hierarchy of items. Root is the one item from which all other items are descended. See root directory or root (/) file system.</td>
</tr>
<tr>
<td><strong>root (/) file system</strong></td>
<td>The top-level file system from which all other file systems stem. The root (/) file system is the base on which all other file systems are mounted, and is never unmounted. The root (/) file system contains the directories and files critical for system operation, such as the kernel, device drivers, and the programs that are used to start (boot) a system.</td>
</tr>
<tr>
<td><strong>root directory</strong></td>
<td>The top-level directory from which all other directories stem.</td>
</tr>
<tr>
<td><strong>rule</strong></td>
<td>A series of values that assigns one or more system attributes to a profile. A rule is used in a JumpStart installation.</td>
</tr>
<tr>
<td><strong>rules file</strong></td>
<td>A text file that contains a rule for each group of systems or single systems that you want to install automatically. Each rule distinguishes a group of systems, based on one or more system attributes. The rules file links each group to a profile, which is a text file that defines how the Oracle Solaris software is to be installed on each system in the group. A rules file is used in a JumpStart installation. See also profile.</td>
</tr>
<tr>
<td><strong>rules.ok file</strong></td>
<td>A generated version of the rules file. The rules.ok file is required by the JumpStart installation software to match a system to a profile. You must use the check script to create the rules.ok file.</td>
</tr>
<tr>
<td><strong>slice</strong></td>
<td>The unit into which the disk space is divided by the software.</td>
</tr>
<tr>
<td><strong>snapshot</strong></td>
<td>A read-only image of a ZFS file system or volume at a given point in time.</td>
</tr>
<tr>
<td><strong>software group</strong></td>
<td>A logical grouping of the Oracle Solaris software (clusters and packages). During an Oracle Solaris installation, you can install one of the following software groups: Core, End User Oracle Solaris Software, Developer Oracle Solaris Software, or Entire Oracle Solaris Software, and for SPARC systems only, Entire Oracle Solaris Software Group Plus OEM Support.</td>
</tr>
<tr>
<td><strong>standalone</strong></td>
<td>A computer that does not require support from any other machine.</td>
</tr>
<tr>
<td><strong>state database</strong></td>
<td>A database that stores information about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a state database replica. The state database tracks the location and status of all known state database replicas.</td>
</tr>
<tr>
<td><strong>state database replica</strong></td>
<td>A copy of a state database. The replica ensures that the data in the database is valid.</td>
</tr>
<tr>
<td><strong>submirror</strong></td>
<td>See RAID-0 volume.</td>
</tr>
<tr>
<td><strong>swap space</strong></td>
<td>A slice or file that temporarily holds the contents of a memory area till it can be reloaded in memory. Also called the /swa or swap volume.</td>
</tr>
<tr>
<td><strong>sysidcfg file</strong></td>
<td>A file in which you specify a set of special system configuration keywords that preconfigure a system.</td>
</tr>
</tbody>
</table>
update
An installation, or to perform an installation, on a system that changes software that is of the same type. Unlike an upgrade, an update might downgrade the system. Unlike an initial installation, software of the same type that is being installed must be present before an update can occur.

upgrade
An installation that merges files with existing files and preserves modifications where possible.

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

upgrade option
An option that is presented by the Oracle Solaris installation program. The upgrade procedure merges the new version of Oracle Solaris with existing files on your disk or disks. An upgrade also saves as many local modifications as possible since the last time Oracle Solaris was installed.

/usr file system
A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large /usr file system with a server rather than maintaining a local copy minimizes the overall disk space that is required to install and run the Oracle Solaris software on a system.

/var file system
A file system or directory (on standalone systems) that contains system files that are likely to change or grow over the life of the system. These files include system logs, vi files, mail files, and UUCP files.

virtual device
A logical device in a ZFS pool, which can be a physical device, a file, or a collection of devices.

volume
A group of physical slices or other volumes that appear to the system as a single logical device. A volume is functionally identical to a physical disk for the purposes of an application or file system.

In some command-line utilities, a volume is called a metadevice. Volume is also called pseudo device or virtual device in standard UNIX terms.

Volume Manager
A program that provides a mechanism to administer and obtain access to the data on DVD-ROMs, CD-ROMs, and diskettes.

ZFS
A file system using storage pools to manage physical storage.

zone
See non-global zone
Index

Numbers and Symbols

#  
in profiles, 32  
in rules files, 29  
&& (ampersands) rule field, 30  
( / ) file systems  
value set by JumpStart, 143  
= (equal sign) in profile field, 52  
! (exclamation mark) rule field, 29

A

add_install_client command, JumpStart directory  
access, 23  
adding  
clusters when upgrading, 115  
packages and patches with a finish script, 55  
packages from software groups, 132  
rules to rules file, 29  
alternative installation programs, 65  
ampersands (&&) rule field, 30  
AND rule field, 30  
any  
probe keyword, description and values, 148  
rule keyword, description and values, 97,147  
arch probe keyword, 147  
arch rule keyword, 97,147  
arhive  
JumpStart profile example, 37,38,39  
keywords, JumpStart, 103–109  
archive_location keyword, 103–109  

c auto_install_sample directory  
check script, 49,70  
copying files to JumpStart directory, 22,26,27

B

-b option of setup_install_server command, 90  
backslash in rules files, 29  
backup_media keyword, 109–110  
begin.log file, 52  
begin rule field, description, 30  

begin scripts  
creating derived profiles with, 52,53  
overview, 51  
permissions, 52  
rule field, 30  
site-specific installation programs, 65  
tracking installation duration, 53  
boot_device keyword, 111  
bootenv createbe keyword, 112  
booting  
creating a profile diskette, 26  
installing with GRUB, 82  
with GRUB, command reference, 85  

Bourne shell scripts in rule fields, 30

C

-c option  
add_install_client command, 93,94  
pfinstall command, 47
Index

changing directories
  image of Oracle Solaris SPARC software on local disk, 26
to image of Oracle Solaris software on local disk, 22
to image of Oracle Solaris x86 based software on local disk, 27
to JumpStart directory, 49, 70
check script
  custom_probes file validation, 70, 71
custom_probes_ok file creation, 70
derived profiles and, 53
rules file validation, 49, 50, 71
rules.ok file creation, 49
testing rules, 49, 71
client_arch keyword, 113
client_root_profile keyword, 113
cluster profile keyword
  description and values, 114–115, 115
  examples, 33
comments
  in profiles, 32
  in rules files, 29
configuring, creating disk configuration files, 59
coping
  JumpStart directory files, 55
  JumpStart installation files, 22, 26, 27
Core Oracle Solaris Software Group, 114–115
CPUs (processors)
  probe keywords, 147
  rule keywords, 97, 147
creating
  custom_probes.ok file, 70
disk configuration files, 59
JumpStart directory, on server, 21
local file systems, 120–122
profiles
  derived, 52
  description, 32
RAID-1 volumes, 122–124
rules file, 28
rules.ok file, 49, 70
.cshrc file, 57
custom_probes file
  naming, 68

custom_probes file (Continued)
  requirements, 68
testing custom_probes, 71
  validating by using check, 70, 71
custom_probes.ok file
  creating, 70
description, 70

D
defaults
  derived profile name, 53
partitioning
  designating disks, 145
  excluding disks, 116
software group installed, 115
deleting, clusters when upgrading, 115
derived profiles, 52, 53
Developer Oracle Solaris Software Group, 114–115
profile example, 33
dfstab file, 91
directories
changing
to image of Oracle Solaris software on local disk, 22
to image of Oracle Solaris SPARC software on local disk, 26
to image of Oracle Solaris x86 based software on local disk, 27
to JumpStart directory, 49, 70
JumpStart
  adding files, 55
  copying files, 55
  copying installation files, 22, 26, 27
  creating directory, 91
  creating for systems, 24
  permissions, 21, 24
  rules file example, 29
  sharing directory, 91
disk configuration files
  creating
    x86 based systems, 62, 64
description, 45

166 Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations • January 2013
diskettes
  JumpStart directory access, 23
x86: JumpStart directory, 24
diskless clients
  platforms, 113
  swap space, 113
disks probe keyword, description and values, 147
disksize rule keyword, description and values, 98, 147
display
  tip line connection requirements, 78, 82
domainname probe keyword, 147
domainname rule keyword, 98, 147
domains
  probe keyword, 147
  rule keyword, 98, 147
don't use profile keyword, 116, 145

files and file systems (Continued)
mounting remote file systems, 119–120
filesys keyword, 120–122, 122–124
filesys profile keyword
description and values, 119–120
examples, 33
finish .log file, 54
finish rule field, description, 30
finish scripts
  adding packages and patches, 55
  customizing the root environment, 57
  rule field, 30
  tracking installation duration, 53

G
geo keyword, 124
getfile: RPC failed: error 5: RPC Timed out message, 24
GRUB based booting
  command reference, 85
  creating a profile diskette, 26
  installing, 82

H
hard disks
  mounting, 119–120
partitioning
  designating for partitioning default, 145
  examples, 33
  excluding for partitioning default, 116
  profile keyword, 137
  rootdisk values, 143
  size
    probe keywords, 147, 148
    root space, 113
    rule keywords, 98, 101, 147, 148
  swap space
    diskless client, 113
    maximum size, 114
    profile examples, 17, 33
  hostaddress probe keyword, 147
  hostaddress rule keyword, 98, 147

Index

JumpStart installation, examples (Continued)
  marketing_profile creation, 92
  marketing systems setup, 90, 94
  networked, 16
  nonnetworked, 15
  RAID-1 volume profiles, 40
  rules file editing, 92, 93
  site setup, 87, 88
  standalone system, 15
  WAN boot installation profile, 37
  optional features, 51
    begin scripts, 51, 53
    finish scripts, 54
    overview, 51
    site-specific installation programs, 65
    overview, 17
    preparing, 17, 50
    profile keywords, 102
    tip line connection requirements, 78, 82

JumpStart directory
  adding files with finish scripts, 55
  copying files
    installation files, 22, 26, 27
    using finish scripts, 55
  creating
    diskette for SPARC based systems, 25
    diskette for x86 based systems, 24, 27
    example, 91
    server, 21
  permissions, 21, 24
  rules file example, 29
  sharing, 21, 91

JumpStart installation
  description, 17
  examples, 95
    booting and installing, 95
    check script, 93
    eng_profile creation, 91
    engineering systems setup, 93
    Flash Archive profile, 37, 38, 39
    JumpStart directory, 91

karch probe keyword, 147
karch rule keyword, 99, 147
keywords
  Flash Archive archives, JumpStart, 103–109
  probe, 67
  limited for ZFS, 150
locale keyword, 129
log files
  begin script output, 52
  finish script output, 54
  logical AND rule field, 30
marketing_profile example, 92
matching
  derived profiles, 53
  order for rules, 31, 77, 82
  rootdisk values, 143
memory
  probe keyword, 147
  rule keyword, 99, 147
  swap space size and, 114
memsiz probe keyword, description and values, 147
memsiz rule keyword, description and values, 99, 147
metadb profile keyword, 129–130
microprocessors
  probe keywords, 147
  rule keywords, 97, 147
model probe keyword, description and values, 147
model rule keyword, description and values, 100, 147
mounting
  begin script caution, 52
  by Solaris installation, 54
  remote file systems, 119–120
multiple lines in rules files, 29

N
names/naming
  custom probes file, 68
  derived profile names, 53
  host name, 98, 147
  rules file, 29
  system model names, 100, 147
network installation, JumpStart installation,
  example, 16
network number, 100, 148
network probe keyword, description and values, 148
network rule keyword, description and values, 100, 148
no_master_check keyword, 130
noneuclidean profile keyword, 131

O
Oracle Solaris software
  groups, (Continued)
    upgrading, 115
  release or version
    installed probe keyword, 147
    installed rule keyword, 99, 147
    osname probe keyword, 148
    osname rule keyword, 100, 148
    osname probe keyword, 148
    osname rule keyword, 100, 148
  output files
    begin script log, 52
    finish script log, 54

P
  -p option of check script, 49, 70
packages
  adding
    with a finish script, 55
    with chroot, 55
  administration file, 51
partitioning
  examples, 33
  excluding disks, 116
  fdisk partitions, 33, 116–119
  profile keyword, 137, 145
  partitioning keyword, 137
patches
  adding
    with a finish script, 55
    with chroot, 55
paths, check script, 49, 70
permissions
  begin scripts, 52
  finish scripts, 54
  JumpStart directory, 21, 24
pfinstall command, 44
platforms
  diskless client, 113
  matching system attributes and profiles, 31, 77, 82
  probe keywords, 147
  rule keywords, 99, 147
  system model names, 100, 147
preparing for installation, with JumpStart, 17, 50
profiles (Continued)
  rule field, 30
  testing, 48
prvtoc command
  SPARC: creating disk configuration file, 60
  x86: disk configuration file creation, 62, 64

R
  -r option of check script, 49, 71
Reduced Network Support Software Group, 114–115
release of Oracle Solaris software
  installed probe keyword, 147
  installed rule keyword, 99, 147
  osname probe keyword, 148
  osname rule keyword, 100, 148
remote file systems, mounting, 119–120
requirements
  custom_probes file, 68
  profiles, 29, 32
root (/) file systems, profile example, 17
root_device keyword, 143
root environment, customizing with a finish script, 57
rootdisk
  definition, 143
  slice value for filesys, 120
  value set by JumpStart, 143
RPC failed: error 5: RPC Timed out message, 24
RPC Timed out message, 24
rule_keyword rule field, 29
rule keywords, 97
  any, description and values, 97, 147
  arch, 97, 147
  disksize, description and values, 98, 147
  domainname, 98, 147
  hostaddress, 98, 147
  hostname, 97–101, 147
  installed, description and values, 99, 147
  karch, 99, 147
  memsize, 99, 147
  model, 100, 147
  network, 100, 148
  osname, 100, 148
  probe, 101
rule keywords (Continued)
  totaldisk, 101, 148
  rule_value rule field, 29
rules
  derived profiles, 52, 53
  examples, 31
  field descriptions, 29, 30
  matching order, 31, 77, 82
  multiple line rules, 29
  rootdisk matching rules, 143
  syntax, 29
  testing validity, 49, 71
rules file
  adding rules, 29
  comments in, 29
  creating, 28
  description, 28
  example, 29
  JumpStart example, 92, 93
  multiple line rules, 29
  naming, 29
  syntax, 29
  testing rules, 49
  validating by using check, 50
  derived profiles and, 53
  JumpStart example, 93
rules.ok file
  creating, 49
  description, 49
rules.ok file, matching order for rules, 31
rules.ok file
  matching order for rules, 77, 82
S
  -s option of add_install_client command, 94
scripts
  begin scripts, 51, 53, 65
  Bourne shell scripts in rule fields, 30
  finish scripts, 54, 65
servers
  JumpStart directory creation, 21
  root space, 113
share command, sharing JumpStart directory, 91
shareall command, 22, 91
sharing JumpStart directory, 91
SI_PROFILE environment variable, 53
site-specific installation programs, 65
size
  hard disk
    probe keywords, 147, 148
    root space, 113
    rule keywords, 98, 101, 147, 148
    memory, 99, 147
  swap space
    diskless client, 113
    maximum size, 114
    profile examples, 17
  tip line connection display dimensions, 78, 82
slices
  probe keyword, 147
  profile examples, 33
  rule keyword, 99, 147
software groups
  for profiles, 114–115
  profile examples, 33
  upgrading, 115
standalone systems
  JumpStart installation example, 15
  profile examples, 33
starting, check script, 49, 50
stty command, 78, 82
SUNWCall group, 114–115
SUNWCprog group, 114–115
SUNWCreq group, 114–115
SUNWCrnet group, 114–115
SUNWCuser group, 114–115
SUNWCXall group, 114–115
swap file systems
  diskless client swap space, 113
  memory size and, 114
  profile examples, 17
  size determination, 114
system_type profile keyword
  description and values, 144
  examples, 33
  testing
    profiles, 44, 48
    validating custom_probes files
      testing custom_probes, 71
      using check, 70
    validating rules files
      derived profiles and, 53
      JumpStart example, 93
      testing rules, 49
      using check, 49, 50, 71
  tip line connection display requirements, 82
  tip line connection requirements, 78
  totaldisk probe keyword, 114
  totaldisk rule keyword, 101, 148
upgrade
  profile keywords, 115, 126, 136
  usedisk profile keyword, description and values, 145
validating
  custom_probes file
    testing, 71
    using check, 71
  rules files
    derived profiles and, 53
    JumpStart example, 93
    testing rules, 49
    using check, 49, 50, 71
/var/sadm/system/logs/begin.log file, 52
/var/sadm/system/logs/finish.log file, 54
variables
  SI_PROFILE, 53
  SYS_MEMSIZE, 47
version of Oracle Solaris software
  installed probe keyword, 147
  installed rule keyword, 99, 147
  osname probe keyword, 148
  osname rule keyword, 100, 148
volcheck command, 25, 27

W
wrapping lines in rules files, 29

Z
ZFS
  keywords, description, 151
  limitations, 150
  overview and planning, 150
  profile examples, 154
  profile keywords
    quick reference, 102