Oracle Solaris 11 System Administration

Student Guide - Volume I

D72896GC21
Edition 2.1
August 2012
D78842

Authors

Anies Rahman

Tammy Shannon

Technical Contributors and Reviewers

Mike Carew

Sreedhar Chalamalasetti

Susan Chang

Mary Ding

Alta Elstad

Al Flournoy

Glynn Foster

Mike Gerdts

Dave Giroux

Tetsuya Harada

Kristi Herd

Darren Kenny

David Laudon

Rosemary Martinak

Dave Maxwell

Dermot McCluskey

Kristi McNeill

Ronan O'Connor

John Powell

Brock Pytlik

Eric Siglin

Enzo Silva

Sue Sohn

Karen Tung

Sean Wilcox

Albert White

Oracle Solaris Documentation Team

Editor

Rashmi Rajagopal

Publishers

Michael Sebastian

Jayanthy Keshavamurthy

Contents

1 Introduction

Overview

- Course goals
- Course agenda
- Introductions
- Your learning center
- Your lab environment

Course Goals

- Enable you to perform basic Oracle Solaris 11 system administration tasks successfully and efficiently
- Present tasks that cover the full spectrum of system administrative responsibilities:
  - OS installation
  - Package management
  - Network, data storage, zones, and user administration
  - Services and process management
  - System monitoring and troubleshooting
- Provide numerous and meaningful practice opportunities

This course is designed to teach system administrators who are new to Oracle Solaris 11 how to perform basic system administration tasks using the operating system.

The goals of this course are to:

- Provide you with the skills you need to perform basic system administration tasks in Oracle Solaris 11 successfully and efficiently. You should be able to contribute immediately when you return to the job.
- Present tasks that cover the full range of system administrative responsibilities, to include installing the OS, managing software packages, and administering networks, data storage, zones, and users. Managing services and processes as well as system monitoring and basic troubleshooting are also covered in this course.
- Provide meaningful practice opportunities in each lesson to help you learn the “why” and “how” of each task that you perform on the job

Course Agenda: Day 1

- Lesson 1: Introduction
- Lesson 2: Installing Oracle Solaris 11 by Using an Interactive Installer
  - Planning for an Oracle Solaris 11 OS Installation
  - Installing Oracle Solaris 11 Using an Interactive Installer
  - Verifying the Operating System Installation

The *Oracle Solaris 11 System Administration* course consists of five days of lectures and practice activities.

Lesson 2 covers installation of the Oracle Solaris 11 operating system by using an interactive installer. You learn about how to plan for the installation and then how to perform the installation. You also look at how to verify the operating system installation and establish a system baseline.

As part of Lesson 2, you practice performing an interactive installation as well as the verification of the installation.

Course Agenda: Day 2

- Lesson 3: Updating and Managing Software Packages
  - Planning for an Oracle Solaris 11 OS Software Update
  - Updating the Oracle Solaris 11 OS Using IPS
  - Administering Boot Environments
- Lesson 4: Administering Services
  - Administering SMF Services
  - Booting and Shutting Down a System
- Lesson 5: Setting Up and Administering Data Storage
  - Planning for Data Storage Management
  - Managing ZFS Storage Pools

In Lesson 3, we discuss planning for a software update, followed by updating the OS by using a technology called the Image Packaging System (IPS). The lesson concludes with a discussion about administering boot environments.

Lesson 4 is on administering services, introducing you to the Service Management Facility (SMF). You also learn how to boot and shut down a system.

You will begin Lesson 5, which covers setting up and administering data storage. You start by planning how to manage data storage and then discuss how to set up the data storage by using ZFS storage pools.

Course Agenda: Day 3

- Lesson 5: Setting Up and Administering Data Storage (continued)
  - Managing ZFS File Systems
  - Using ZFS Snapshots and Clones
- Lesson 6: Administering Oracle Solaris Zones
  - Planning for Oracle Solaris Zones
  - Determining an Oracle Solaris Zone Configuration
- Lesson 7: Administering a Physical Network
  - Planning for Network Management
  - Determining Datalink Availability
  - Configuring a Network Interface
  - Administering a Network Interface
  - Verifying Network Operation

You continue Lesson 5 by learning about how to manage your ZFS file systems. This lesson concludes with a discussion on how to use ZFS snapshots and clones as part of the data storage management.

Lesson 6 is on administering zones. You are introduced to zones and zones concepts. You are then shown how to determine the system's current zone configuration and how to perform basic zone administration tasks, such as logging in to a zone, exiting a zone, shutting down a zone, and starting up a zone.

In Lesson 7, you learn about how to administer a physical network. You look at how to plan for a physical network as well as how to determine a network interface configuration. You are shown how to verify network operations and monitor datalinks.

Course Agenda: Day 4

- Lesson 8: Setting Up and Administering User Accounts
  - Planning for User Administration
  - Setting Up and Maintaining User Accounts
  - Managing User Initialization Files
  - Configuring User Shells and User Disk Quotas
- Lesson 9: Controlling Access to Systems and Files
  - Planning for System and File Access Control
  - Controlling Access to Systems

Lesson 8 covers how to plan for user administration, how to set up and maintain user accounts, how to manage user initialization files, and how to configure user shells and user disk quotas.

Lesson 9 is about controlling access to systems and files. You begin by learning about how to plan to control access to systems and files and then look at how to actually control access to systems.

Course Agenda: Day 5

- Lesson 9: Controlling Access to Systems and Files (continued)
  - Controlling Access to Files
  - Configuring and Using Secure Shell
- Lesson 10: Managing System Processes and Scheduling System Tasks
  - Managing System Processes
  - Scheduling System Tasks
- Lesson 11: Performing Basic System Monitoring and Troubleshooting
  - Monitoring System Logs and Crash and Core Dump Files
  - Performing Basic Troubleshooting on Key System Administration Tasks

Note: Class is from 9:00 AM to 5:00 PM each day. There will be several short breaks throughout the day, plus one hour for lunch.

You continue Lesson 9 by looking at how to control access to files. This lesson concludes with a discussion on how to configure and use Secure Shell.

In Lesson 10, you look at how to manage system processes and schedule system tasks.

In Lesson 11, you are introduced to system logs and crash and core dump files. The lesson concludes by discussing how to perform basic troubleshooting on key system administration tasks, such as software update failures and user account access issues.

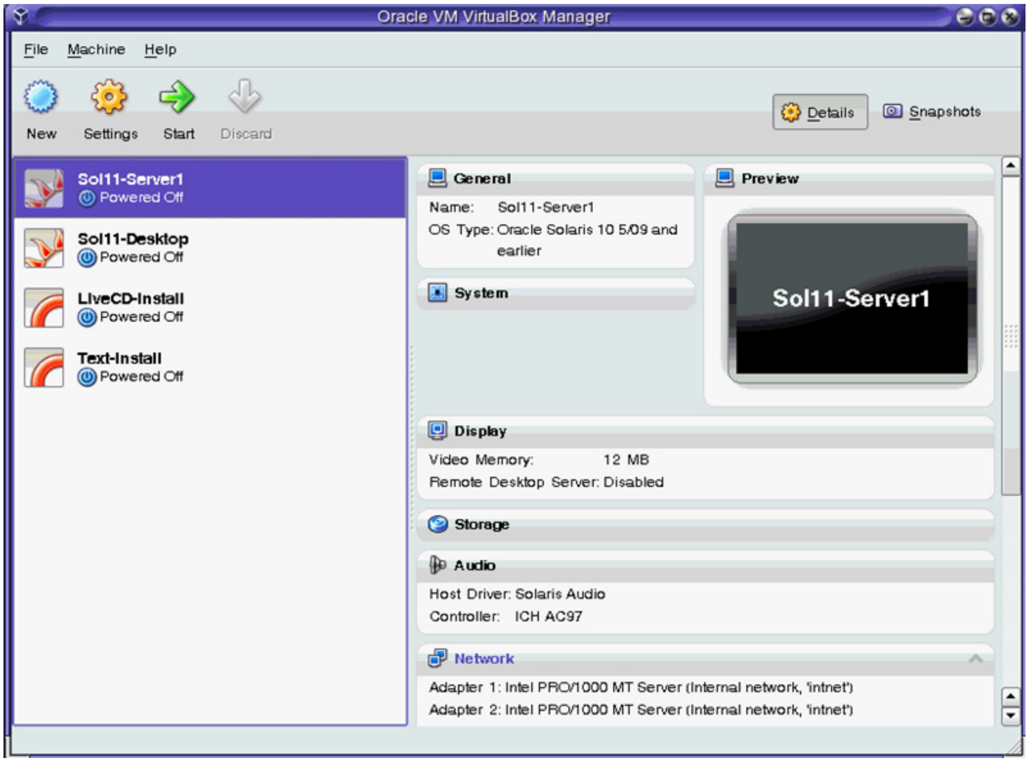
Introductions

- Name
- Company affiliation
- Title, function, and job responsibility
- Experience related to topics presented in this course
- Reasons for enrolling in this course
- Expectations for this course

Your Learning Center

- Logistics
  - Restrooms
  - Break rooms and designated smoking areas
- Cafeterias and restaurants in the area
- Emergency evacuation procedures
- Instructor contact information
- Cell phone usage
- Online course attendance confirmation form

Your Lab Environment



As part of each lesson, you practice—in a lab environment—what you have learned in the lecture. The lab environment we use in this course is based on the Oracle VM VirtualBox virtualization software (an example of which is shown in the slide). VirtualBox is a cross-platform virtualization application. It extends the capabilities of your existing computer so that it can run multiple operating systems (inside multiple virtual machines) at the same time.

Open your *Activity Guide* to the practices for Lesson 1. Your instructor will walk you through the material, and you will have a chance to familiarize yourself with the lab environment configuration and setup.

2

Installing Oracle Solaris 11 by Using an Interactive Installer

Objectives

After completing this lesson, you should be able to:

- Implement a plan for an Oracle Solaris 11 operating system installation
- Install the Oracle Solaris 11 operating system by using an interactive installer
- Verify the operating system installation

This lesson explains how to use an interactive installer to perform an initial installation of the Oracle Solaris 11 operating system.

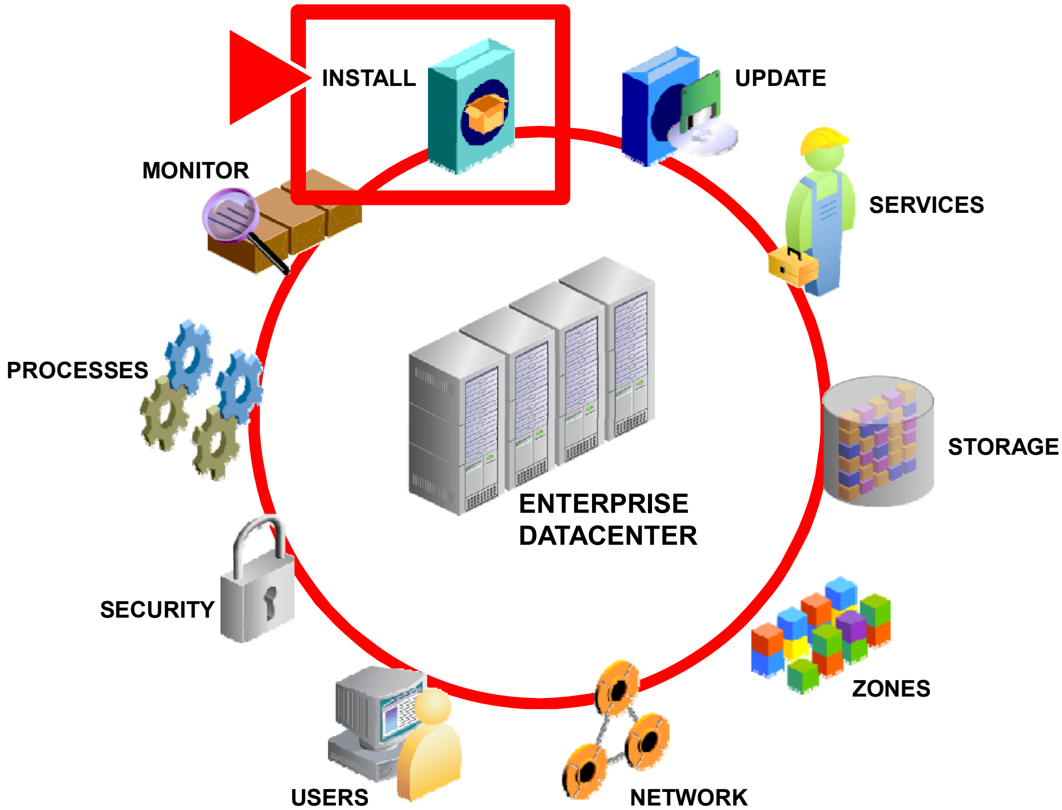
The lesson begins by discussing the importance of planning for the operating system installation. This discussion includes an introduction to the interactive and automated installation options.

Next, you learn about how to install Oracle Solaris 11 by using the interactive installation method.

The lesson concludes by discussing how to verify the operating system installation.

At various points during the lecture, you will have an opportunity to apply what you have learned in a practice environment.

Workflow Orientation



In this course, each of the primary system administration tasks is presented in the context of a workflow. At the start of each lesson, the diagram shown in the slide is used to indicate where you are in the workflow environment (which progresses clockwise around the diagram). You learn about why certain tasks precede or follow other tasks, and the importance of each task as it pertains to the system administrator's job.

As indicated in the diagram, you start (at the top) with installation, followed by how to update and manage software packages, manage services, and administer data storage, zones, and the network. You then learn about how to administer users, control system and file access, and manage processes. Your final tasks are basic system monitoring and troubleshooting.

Lesson Agenda

- Planning for an Oracle Solaris 11 OS Installation
- Installing Oracle Solaris 11 by Using an Interactive Installer
- Verifying the Operating System Installation

Importance of Working from a Plan

Implementing tasks in accordance with a plan guarantees that tasks are:

- Assigned to the appropriate personnel
- Completed as required
- Completed on schedule

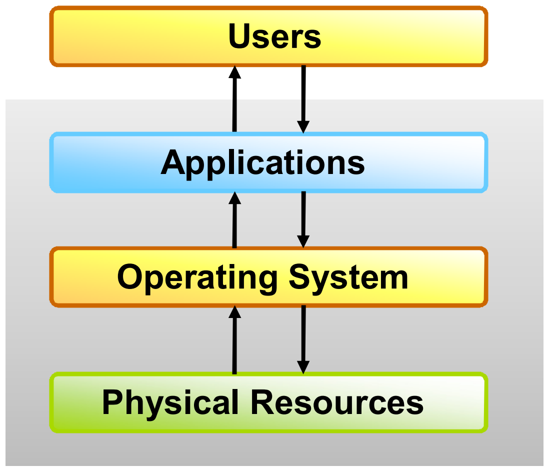
Regardless of the type of task you are performing, but especially for key tasks like installation, you should be executing the task based on a plan. In a large datacenter environment where system administration responsibilities are distributed among multiple administrators, it is even more important that you understand what your responsibilities are, as well as why and when you perform them. As a new system administrator, you will probably be given your direction or plan by a senior person and asked to implement or execute that plan. In some datacenters, the plan is referred to as a *run book*.

In this course, you are given a plan at the start of each major task, and then asked to implement the task as outlined in the plan.

Planning for an Oracle Solaris 11 OS Installation

The operating system:

- Controls and manages physical resources
- Serves as a liaison between a system's users, software applications, and physical resources



To have a functional system, you must install an operating system (OS) on it. The OS controls and manages a system's physical resources (such as the system's hardware components) and serves as a liaison between a system's users, software applications, and physical resources. In short, the OS is "the brain" of the operation.

After it is installed, the OS makes all critical functional components available to users of the system.

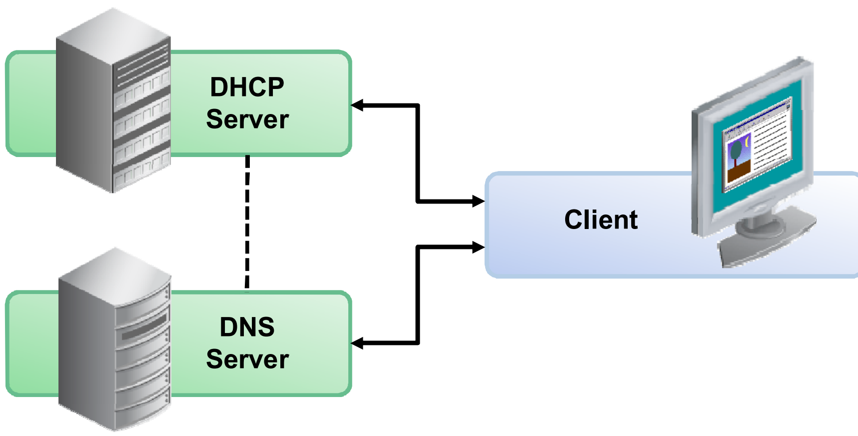
Planning for an Oracle Solaris 11 OS Installation

- Planning is required to make sure that the operating system is:
  - Installed properly
  - Configured to support business needs
- Planning addresses and answers such questions as:
  - How many users will we need to support?
  - What applications will we be running?
  - What type of network will we be using?
  - What are our data storage needs?
  - What are our hardware needs?

Because the operating system is the backbone of datacenter operations and needs to be able to support the needs of the business, it is important that it be installed and configured properly. This requires planning. For example, when will the operating system be installed, on what systems, and how? What testing will be done on the system before moving it into production? What system requirements must be met to install the system? How many users and what type of applications will be supported? What type of network are we using? What data storage needs do we have? What hardware do we need? These types of questions as well as many others must be answered *before* the installation actually occurs.

Identifying Network Configuration Options

- Automatic network configuration
  - Enabled by default during installation
  - Requires DHCP setup



- Manual network configuration

When installing the Oracle Solaris 11 operating system, you have two network configuration options:

- Having the network automatically configured
- Configuring the network manually

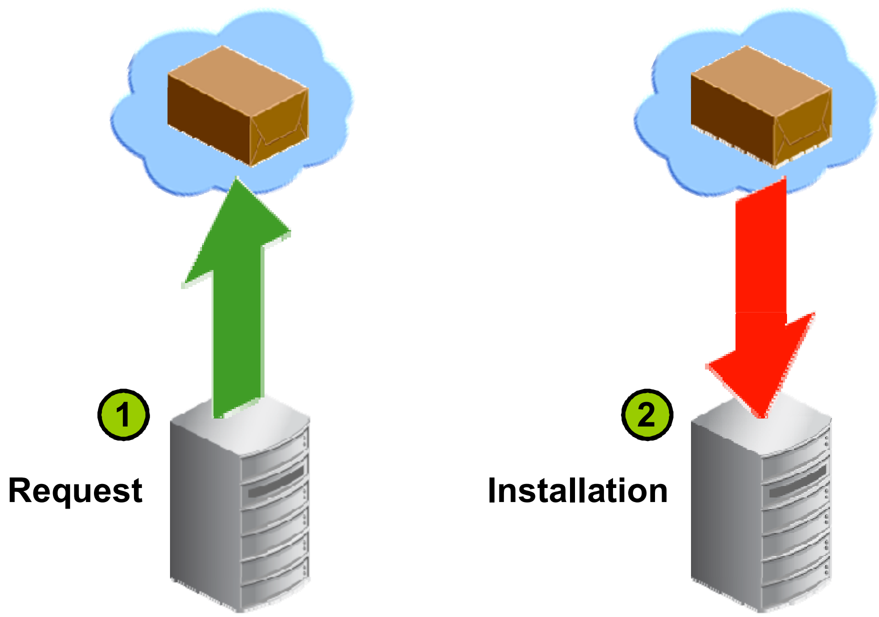
If you want to have the network configured automatically, Oracle Solaris configures the network by using a feature called network auto-magic (NWAM) if the Dynamic Host Configuration Protocol (DHCP) has been set up. DHCP is a facility that dynamically assigns IP addresses and connects the system to a local domain name system (DNS) server, which translates Internet domain names and host names to IP addresses. The DHCP server also provides boot block information.

If you are configuring the network manually, you do not need DHCP.

You will learn more about network configurations, both physical and virtual, in the lesson titled "Administering a Physical Network."

In this lesson, you are shown how to configure the network both automatically and manually.

Installation Process



Oracle installation download website:
http://www.oracle.com/technetwork/server-storage/solaris11/downloads

Planning also helps in specifying the type of system on which you are going to install the OS as well as how the installation will be performed. The Oracle Solaris 11 operating system runs on both x86 and SPARC hardware.

The Oracle Solaris 11 operating system is installed using an ISO image for either x86 or SPARC that has been downloaded from the Oracle installation download website to a server (the site’s URL is provided in the slide).

Methods for Installing an Oracle Solaris 11 Operating System

	Interactive		Automated
	LiveCD GUI	Text Installer	Automated Installer (AI)
SPARC		X	X
x86	X	X	X
Single system	X	X	X
Multiple-client systems			X

You can install Oracle Solaris 11 by using either an interactive or an automated installation. With an interactive installation, you have two options:

- The LiveCD for x86-based systems
- A text installer that can be used on either x86 or SPARC machines

These options are designed for installing the OS on a single system, whereas the automated installation option (commonly known as the Automated Installer, or AI) provides a “hands-free” network installation on a single system or for multiple-client systems. The automated installation also gives administrators the ability to create and manage customized installation profiles for different systems. Regardless of the installation method you choose, installation is easy and fast.

In this course, you learn mainly about how to perform an interactive installation. Automated installation is covered in depth in the *Oracle Solaris 11 Advanced System Administration* course, which is the follow-on to this course.

Differences Between LiveCD GUI and Text Installers

Feature	LiveCD GUI	Text Installer
Packages	Installs desktop-based packages	Installs server-based set of packages
Network configuration	Defaults to automatic network configuration	Allows both automatic and manual configuration of the network
`root` user	The `root` user is always configured as a role.	The `root` user might or might not be a role.
Memory	Requires more memory than text installer	Requires less memory than LiveCD GUI installer

When determining which installer to use, there are several important differences you need to keep in mind:

- The LiveCD GUI installs a desktop-based set of packages, whereas the text installer installs a server-based set of packages.
- The LiveCD GUI installer defaults to automatic network configuration only. The text installer allows you to configure networking.
- The `root` user is always configured as a role in the LiveCD GUI installer. In the text installer, the `root` user might or might not be a role.
- The LiveCD GUI installer requires more memory to run than the text installer.

Identifying Pre-installation Tasks

✓	**Identify system requirements.**
✓	**Identify additional installation considerations.**
✓	**Check device drivers.**

Best practice: Always review installation documentation and release notes carefully before performing an installation.

Installation planning should also address preinstallation tasks, such as identifying system requirements and additional installation considerations for running the installer on your system and ensuring that you have all of the necessary device drivers.

Note: This type of information, as well as step-by-step instructions on how to perform the installation, can be found in the online Solaris documentation on the Oracle Technology Network (OTN) website under Documentation > Oracle Documentation > Systems Software. It is always a best practice to review the documentation—specifically the release notes—as part of your pre-installation tasks.

Identifying System Requirements

Hardware	Requirement
Disk space	Recommended minimum: 13 GB
Memory	Recommended minimum: 1 GB
Architectures	• **X86:** 64-bit only • **SPARC:** Oracle Solaris M-series and T-series systems only

Note: The recommended minimums are subject to change with the final release of the software. See the release notes for final disk space and memory recommendations.

ORACLE

The term "system requirements" refers to disk space, memory, and architectures.

The recommended minimum disk space for an Oracle Solaris 11 installation is 13 GB. The recommended minimum memory requirement is 1 GB.

Note: Both the LiveCD GUI and text installers are capable of running with a minimum of 1 GB of memory. However, the minimum amount of memory that is required varies depending on system specifications. If the GUI installer on the LiveCD ISO image does not work on your system, use the text installer.

As has been discussed, Oracle Solaris 11 runs on both x86 (64-bit only) and SPARC hardware.

Note: SPARC support is available on M-series and T-series systems only. Open Boot PROM (OBP) is required to be at 4.17 or higher. Using the latest firmware is recommended.

Identifying Additional Installation Considerations

- LiveCD ISO image installer is for 64-bit x86 platforms only.
- For SPARC-based systems (M-series and T-series servers only), use the text or automated installer.
- Interactive installers can perform an initial installation on:
  - Whole disk
  - Oracle Solaris x86 partition
  - SPARC slice (text installer)
- **Caution:** The installation overwrites all of the existing data on the targeted disk.

In addition to the system requirements, planning should also address any additional considerations that might affect the installation. For example, with Oracle Solaris 11, the installer on the LiveCD ISO image is for 64-bit x86 platforms *only*. If you need to install the operating system on a SPARC-based system, you need to use the text installer or the automated installer.

The operating system is installed on a system's disk. With an interactive installer, you can perform an initial installation on the whole disk, on an Oracle Solaris x86 partition, or on a SPARC slice if you are using the text installer.

Keep in mind that the installation process overwrites any data that exists on the disk that you have identified or targeted for the installation.

You will learn more about how to format disks later in the course. For now, you will be working with disks that have been formatted for you.

Checking Device Drivers

- Take a few minutes to verify that your system has the appropriate drivers required to manage each of its devices.
- Device drivers enable communication between the operating system and the system's devices.

Before you install the operating system, you might want to verify that your system has the appropriate drivers required to manage each of its devices. Without the right drivers, the operating system cannot communicate with the system's devices. The operating system usually already contains the device drivers that you need, so it should not take you long to perform this check.

Quiz

To configure the network manually during an Oracle Solaris 11 initial installation, you must first have DHCP set up.

a. True

b. False

Answer: b

Quiz

In the text installer, the `root` user ____________.

a. Is always configured as a role
b. Might or might not be configured as a role
c. Is never configured as a role

Answer: b

Quiz

The text installer is used for SPARC-based systems only.

a. True

b. False

Answer: b

Lesson Agenda

- Planning for an Oracle Solaris 11 OS Installation
- Installing Oracle Solaris 11 by Using an Interactive Installer
- Verifying the Operating System Installation

The installation planning activities have been completed. The plan contains all the information needed to install the Oracle Solaris 11 operating system successfully.

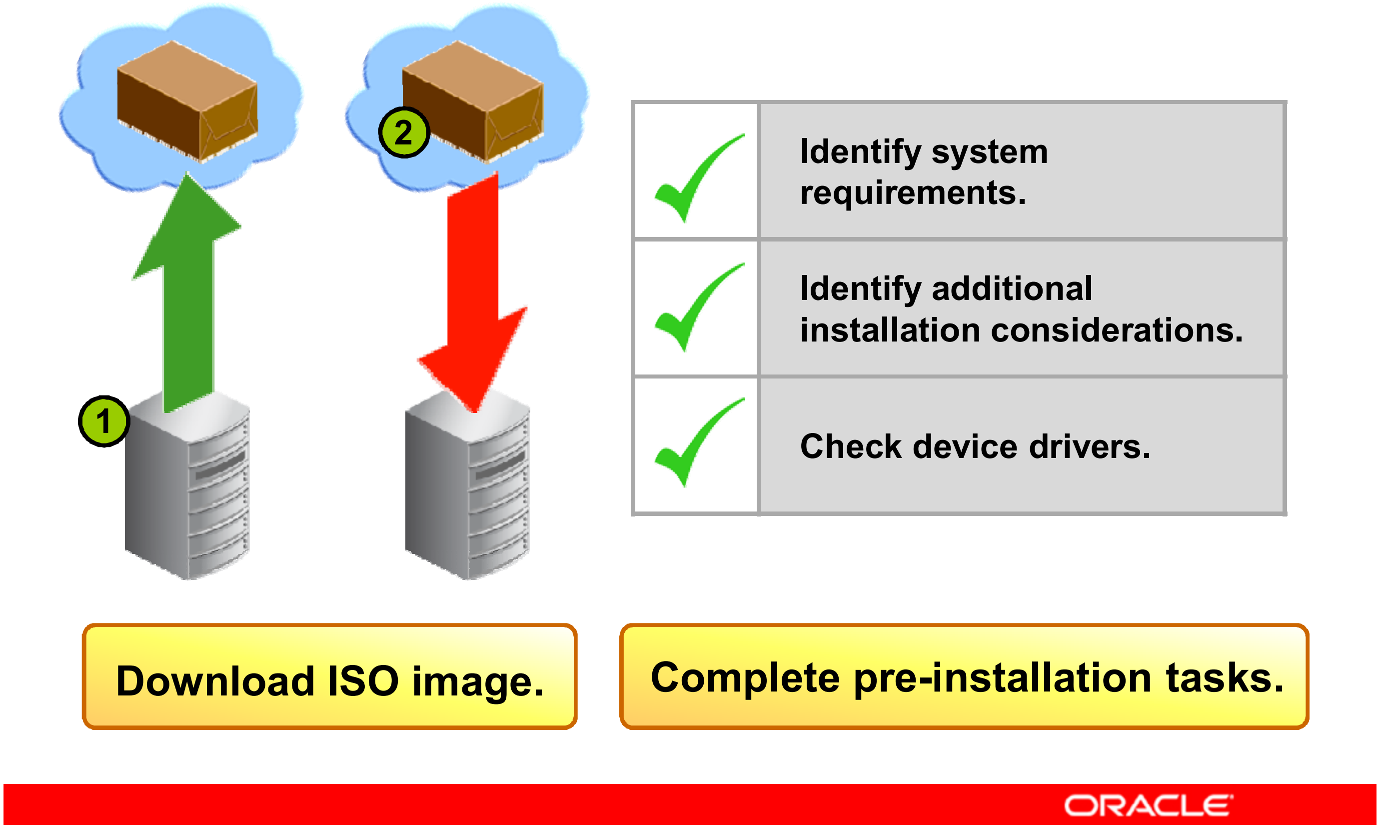
You now look at installation of the operating system on two x86 test machines by using the interactive install options. This section of the lesson provides you with the understanding and skills to complete the hands-on installation tasks in the practices.

Installing Oracle Solaris 11 Using an Interactive Installer

1. Preparing for the installation
2. Performing the installation
3. Verifying the installation
4. Rebooting the system

Installing the operating system consists of four tasks. First, you prepare for the installation and then perform the install. Next, you verify that the install was successful. Then you reboot the system.

Preparing for the Installation



To prepare for the installation, you first download and save the ISO image to your system. Because you are going to be installing the operating system on two x86 test machines by using both the LiveCD and text install, you need to download the following ISO images:

- LiveCD for download for x86
- Text Install for download for x86

You then complete the pre-installation tasks that were identified during planning:

- Identify system requirements (disk space and memory).
- Review additional installation considerations (whole-disk or partition).
- Verify required device drivers.

Performing the Installation

- Installing Oracle Solaris 11 by using the LiveCD
- Installing Oracle Solaris 11 by using the text installer

After you have completed the installation preparation tasks, you are ready to perform the installation. Next, you walk through both the LiveCD and text install installations.

Installing Oracle Solaris 11 by Using the LiveCD with GUI Installer

```
USB keyboard
 1. Albanian                          25. Latin-American
 2. Arabic                            26. Lithuanian
 3. Belarusian                        27. Latvian
 4. Belgian                           28. Macedonian
 5. Brazilian                         29. Malta_UK
 6. Bulgarian                         30. Malta_US
 7. Canadian-Bilingual                31. Norwegian
 8. Croatian                          32. Polish
 9. Czech                             33. Portuguese
10. Danish                            34. Romanian
11. Dutch                             35. Russian
12. Dvorak                            36. Serbia-And-Montenegro
13. Estonian                          37. Slovak
14. Finnish                           38. Slovenian
15. French                            39. Spanish
16. French-Canadian                   40. Swedish
17. Hungarian                         41. Swiss-French
18. German                            42. Swiss-German
19. Greek                             43. Traditional-Chinese
20. Icelandic                         44. TurkishF
21. Italian                           45. TurkishQ
22. Japanese-type6                    46. UK-English
23. Japanese                          47. US-English
24. Korean
To select the keyboard layout, enter a number [default 47]:
```


The Oracle Solaris 11 LiveCD provides a GUI-based interactive installation that walks you through the process of configuring the system for the initial OS installation. The ISO image boots to a full OS with a functional desktop. After you boot the LiveCD with the GUI installer, the first thing you are asked to do is identify the keyboard layout. The default is US-English [47].

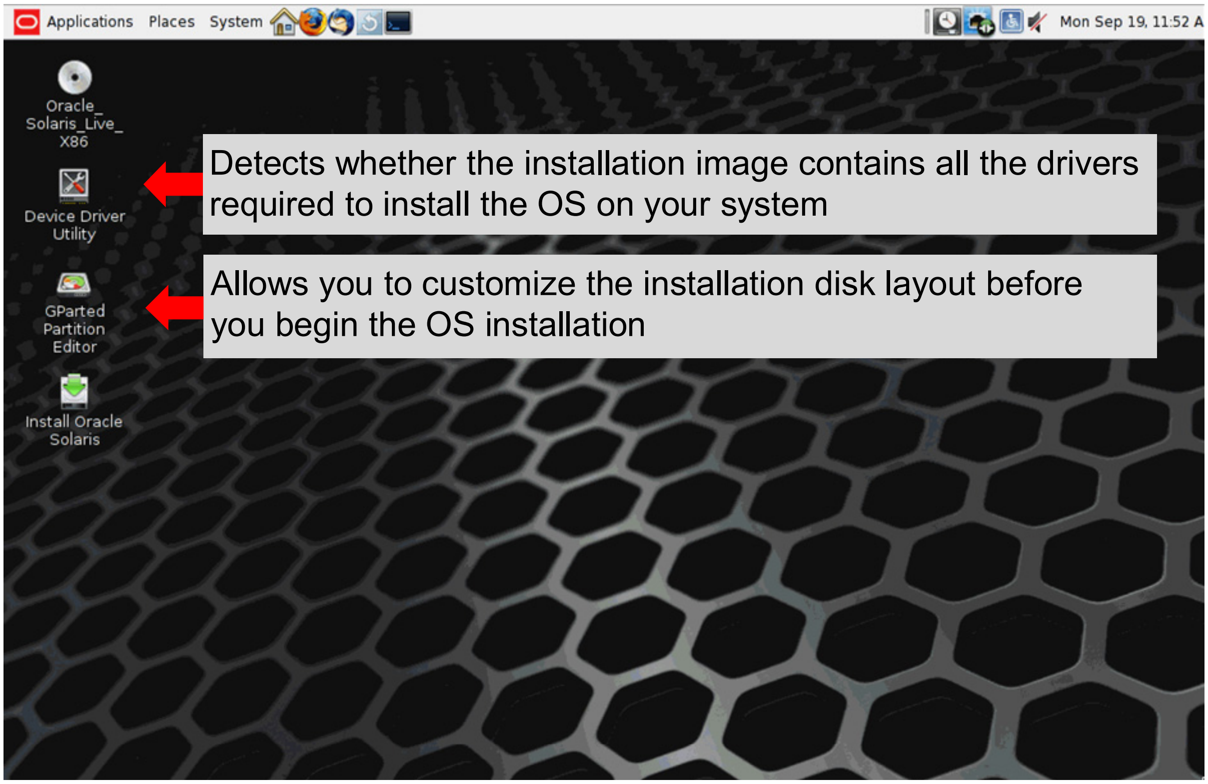
Installing Oracle Solaris 11 by Using the LiveCD with GUI Installer

```
 1. Chinese - Simplified
 2. Chinese - Traditional
 3. English
 4. French
 5. German
 6. Italian
 7. Japanese
 8. Korean
 9. Portuguese - Brazil
10. Spanish
To select the language you wish to use, enter a number [default is 3]:
```

ORACLE

After selecting the keyboard layout, you are prompted to select the language you want to use. Again, English is the default [3]. From this point, the installer configures the system devices and then launches the GUI interface.

Introducing the LiveCD Desktop

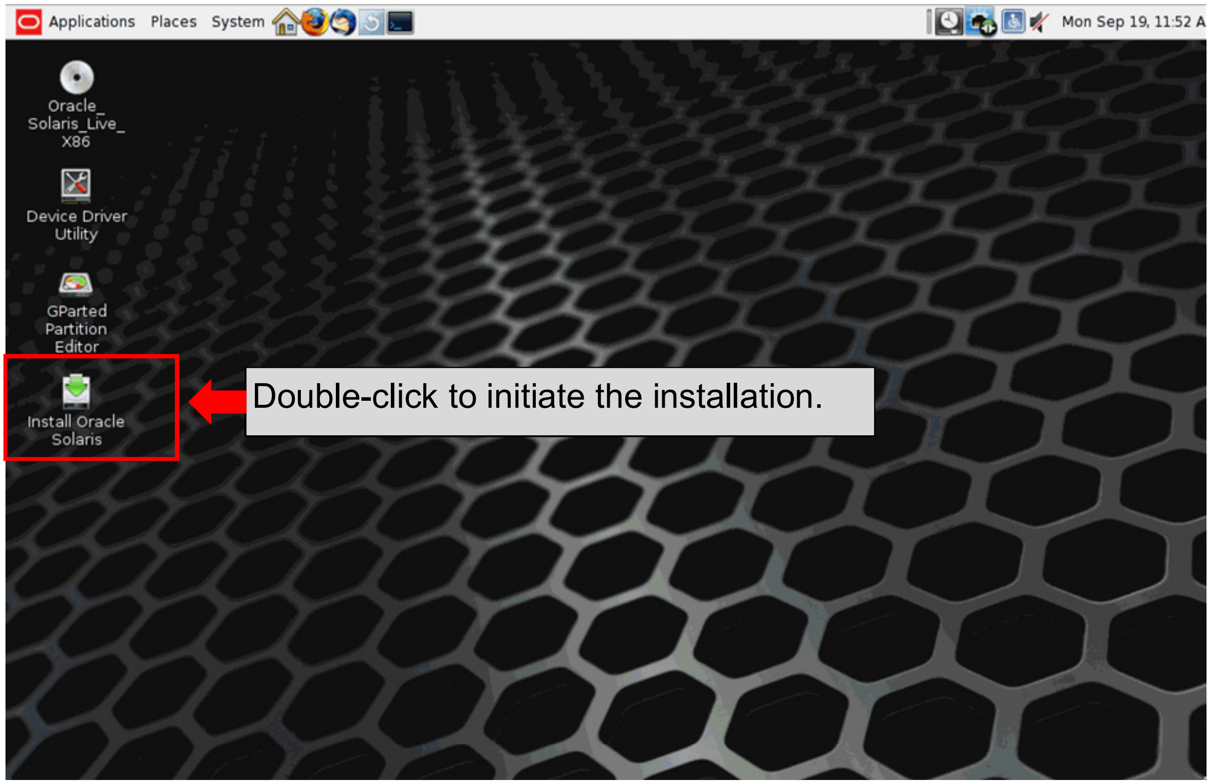


The GUI interface provides a full desktop operating environment (as shown in this slide). The LiveCD provides additional tools to assist you in your installation, such as the Device Driver Utility and a partition editor.

When you boot your system from the LiveCD ISO image, and the desktop is displayed, the Device Driver Utility automatically launches and begins searching for missing device drivers. If the utility locates any such devices, a notification is displayed on the desktop.

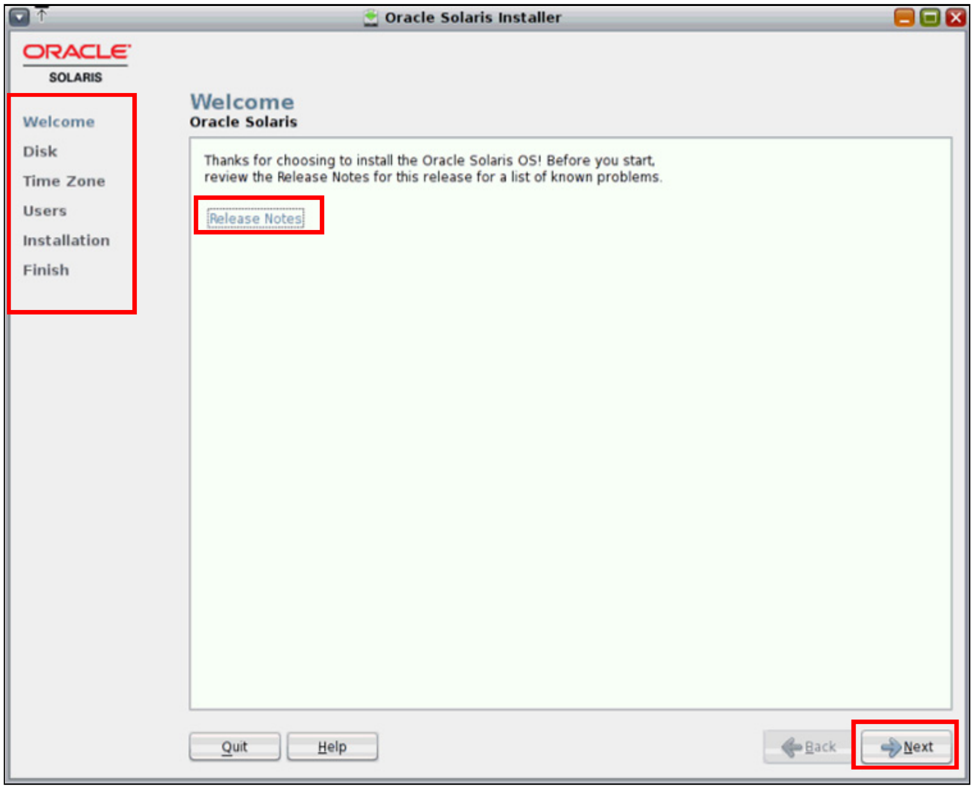
You also have the option of manually launching the Device Driver Utility by double-clicking the icon or selecting Applications > System > Device Driver Utility.

Initiating the Installation with LiveCD



To initiate the installation, double-click the Install Oracle Solaris icon.

Welcome Screen



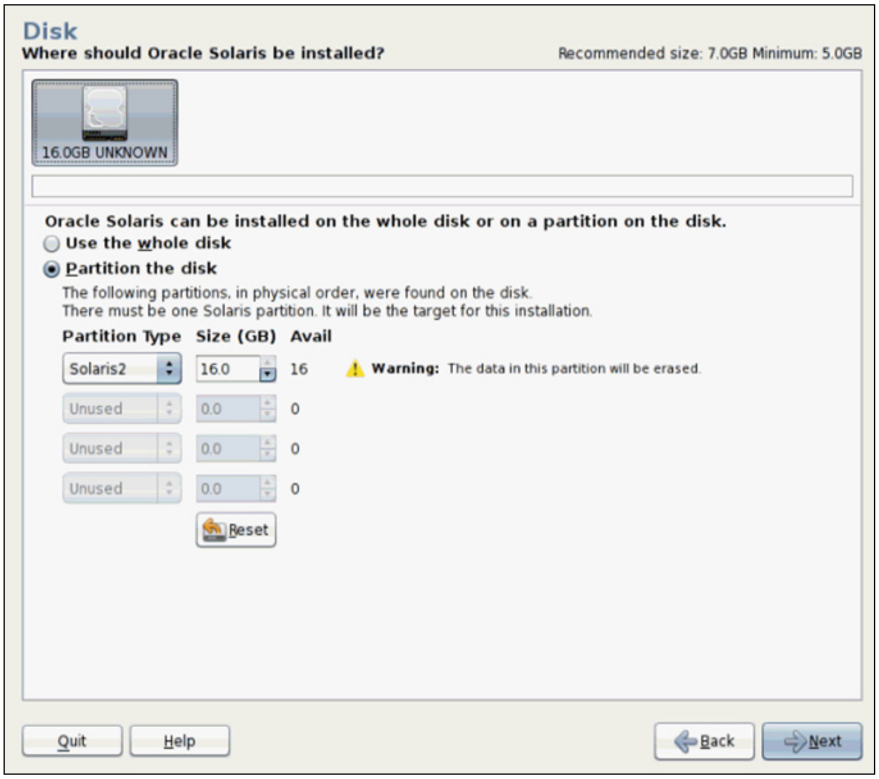
The first screen you see is the Welcome screen. From this screen, you can review the release notes (if you haven't already done so).

The list of items on the left of the Welcome screen highlights the steps you take to complete the installation. You begin the installation by providing configuration data for the disk, time zone, and users. The information you provide should be based on the installation plan you were given to follow.

After you have completed the configuration data screens, the actual installation begins. After the installation is completed, you are done.

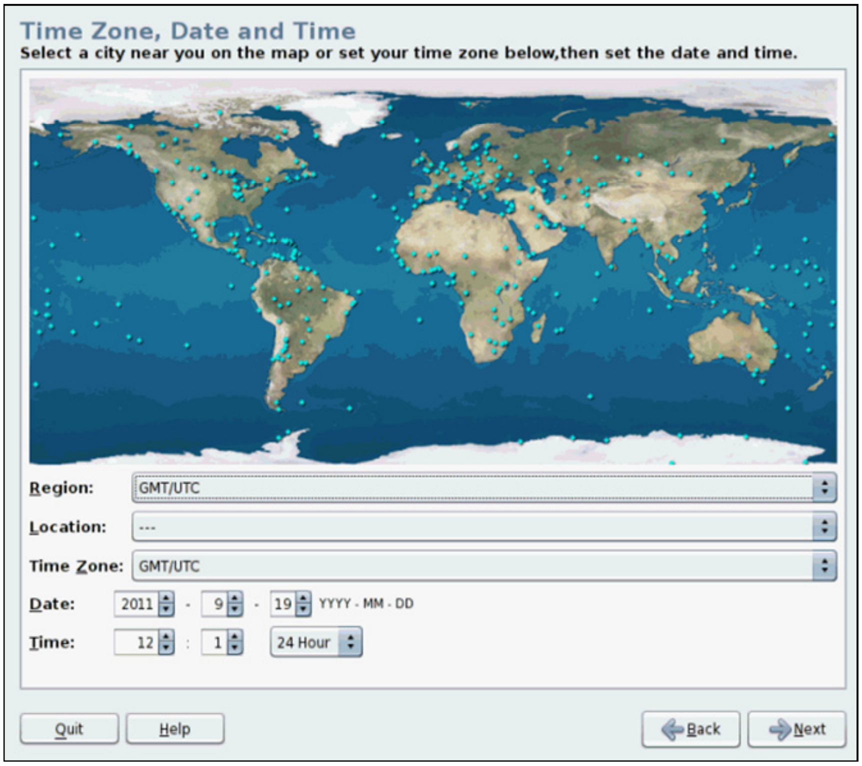
You now walk through each of the installation screens to be introduced to them. To advance to the Disk screen, click Next.

Selecting a Disk



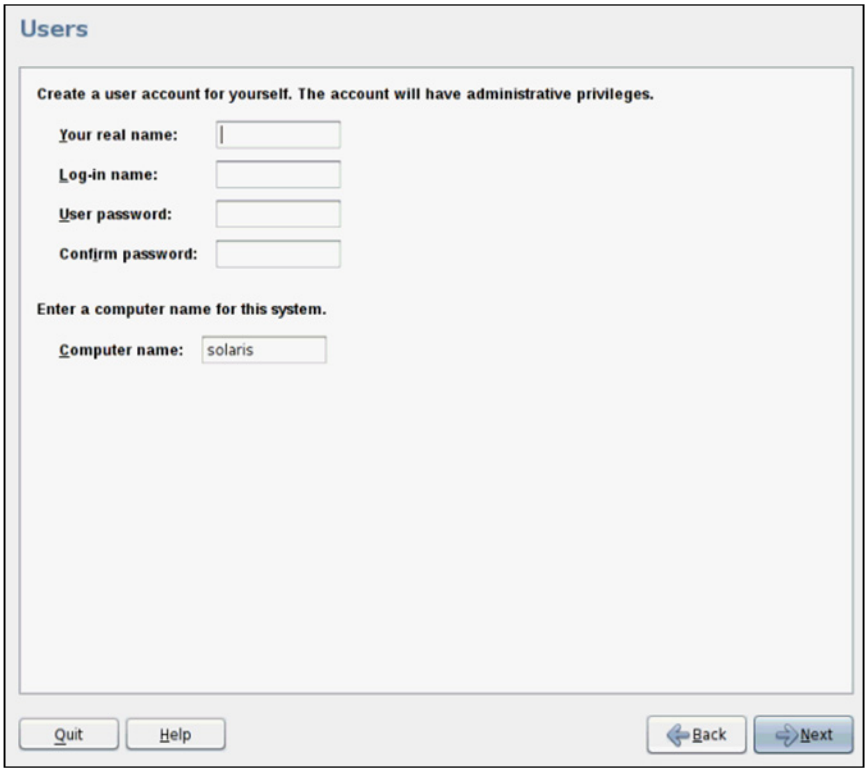
From the Welcome screen, you are taken to the Disk screen, where you are prompted to select where you want Oracle Solaris to be installed. You can select to use a whole disk or to partition the disk, in which case you need to select the partition type and size.

Setting the Time Zone, Date, and Time



The next screen that appears is “Time Zone, Date and Time.” On this page you can select the region, location, and time zone that are appropriate to your installation. You can also set the date and time. To continue, click Next.

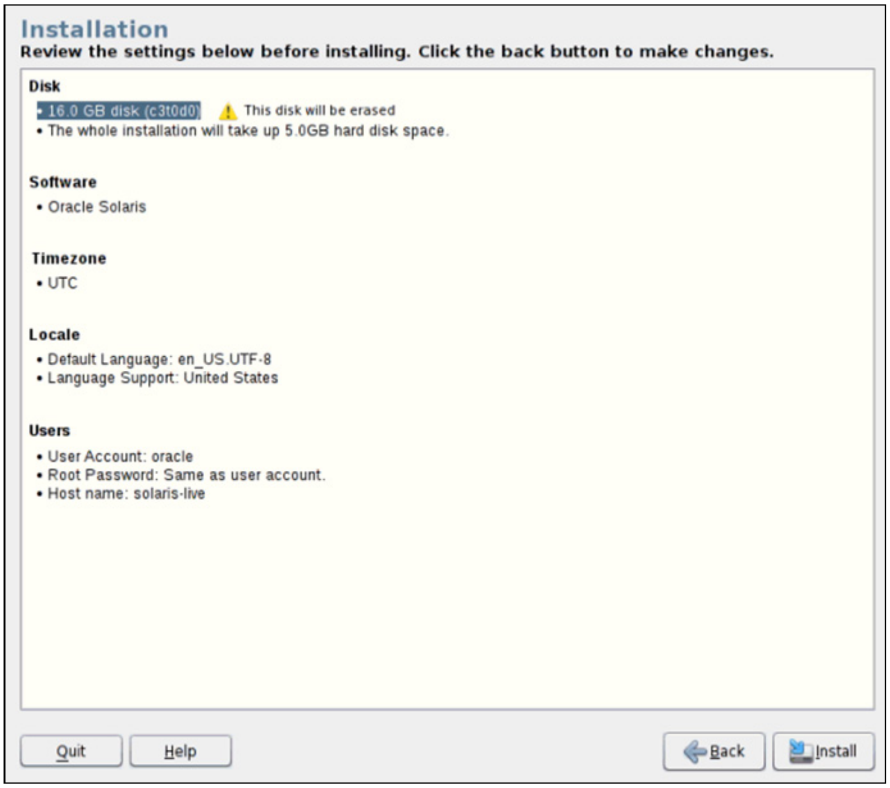
Providing User Information



Next is the Users screen, where you enter your user information, including your real name, your login name, and your user password. You are also asked to provide a computer name, which is also referred to as the host name or node name. To continue, click Next.

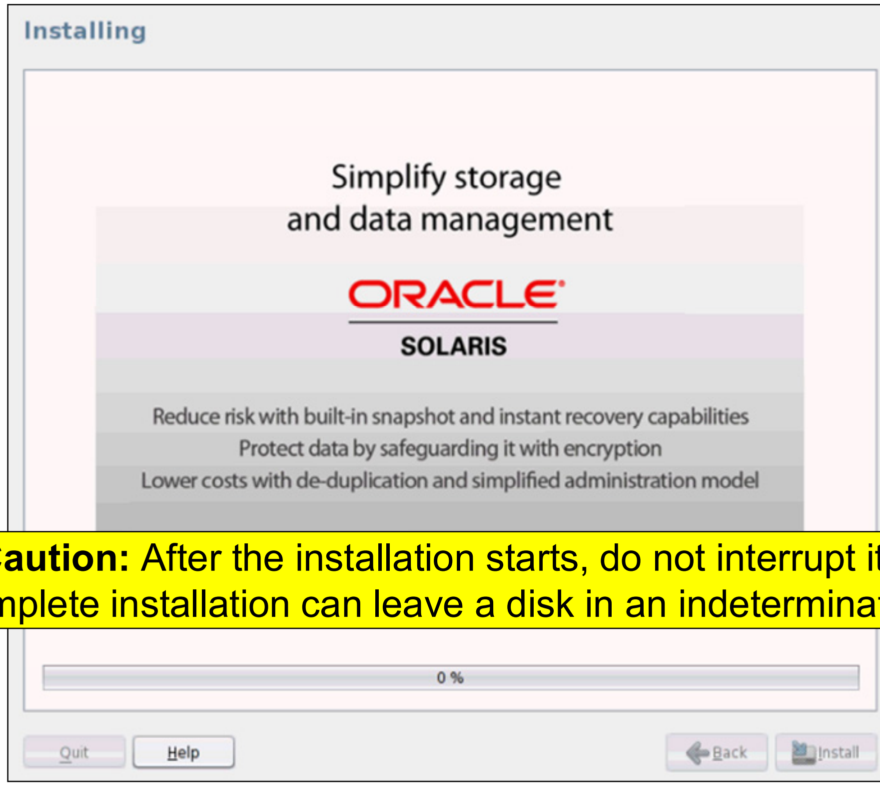
Note: The first user configured is given the `root` role.

Reviewing Installation Specifications



After you have completed the configuration data, you see the Installation summary screen. Review the information carefully to make sure it is accurate before you start the installation. You can go back and make changes if necessary.

Monitoring the Installation



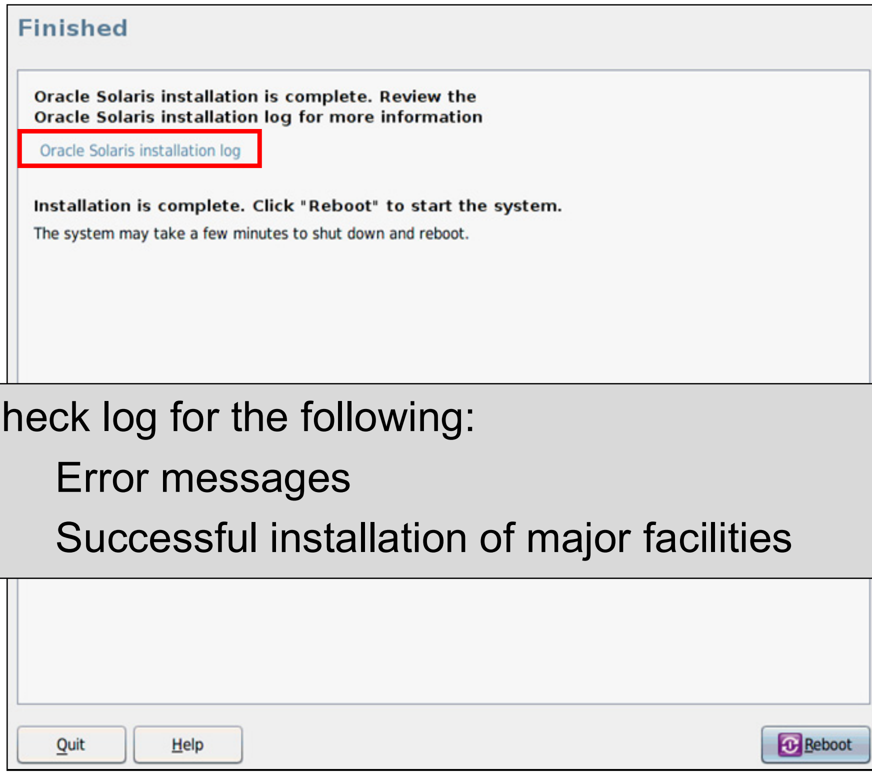
Caution: After the installation starts, do not interrupt it.
An incomplete installation can leave a disk in an indeterminate state.

The Installation screen enables you to monitor the progress of the installation. The installation takes about 15 to 20 minutes to complete.

Keep in mind that after the installation starts, you do not want to interrupt it. Interrupting an installation can leave a disk in an indeterminate state.

Note: In your Virtual Box training environment, the install might take up to 50 minutes to complete.

Verifying the Installation



Check log for the following:

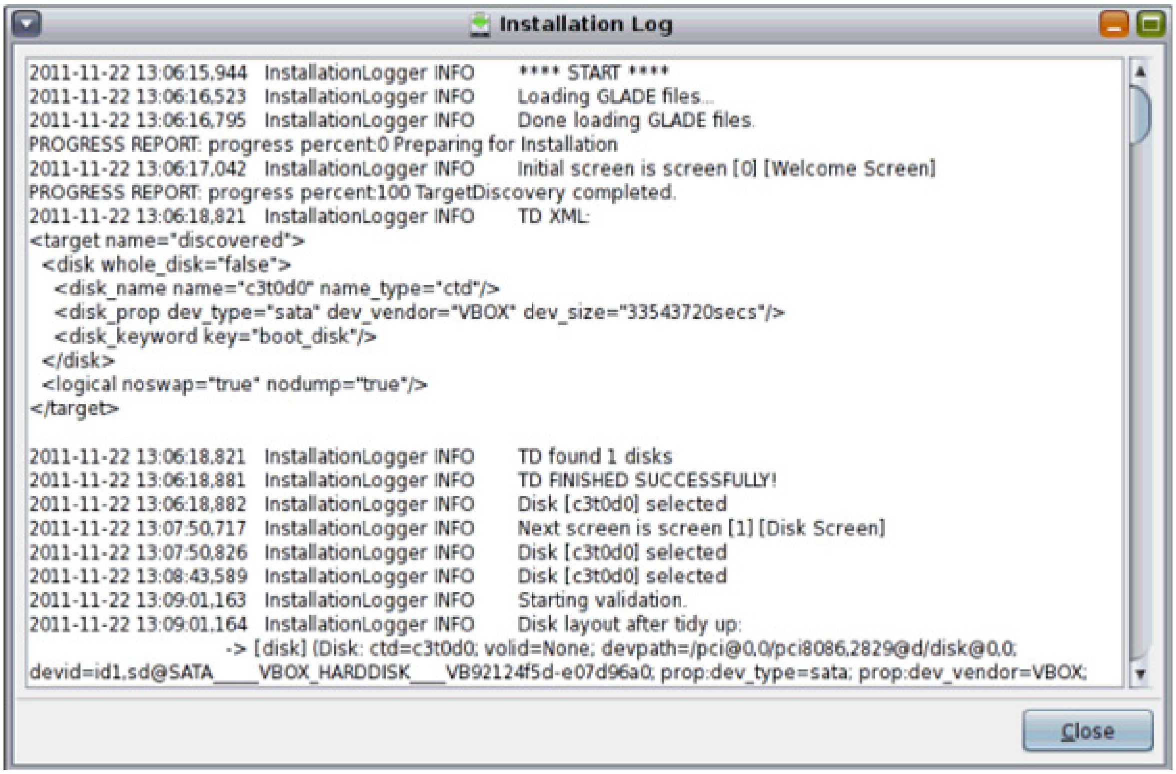
- Error messages
- Successful installation of major facilities

When the installation concludes, you see the Finished screen, which provides you with access to the installation log and an opportunity to verify that:

- No errors occurred during installation
- Major facilities were successfully installed

To access the log, click the “Oracle Solaris installation log” link. A separate dialog box then appears with the log contents.

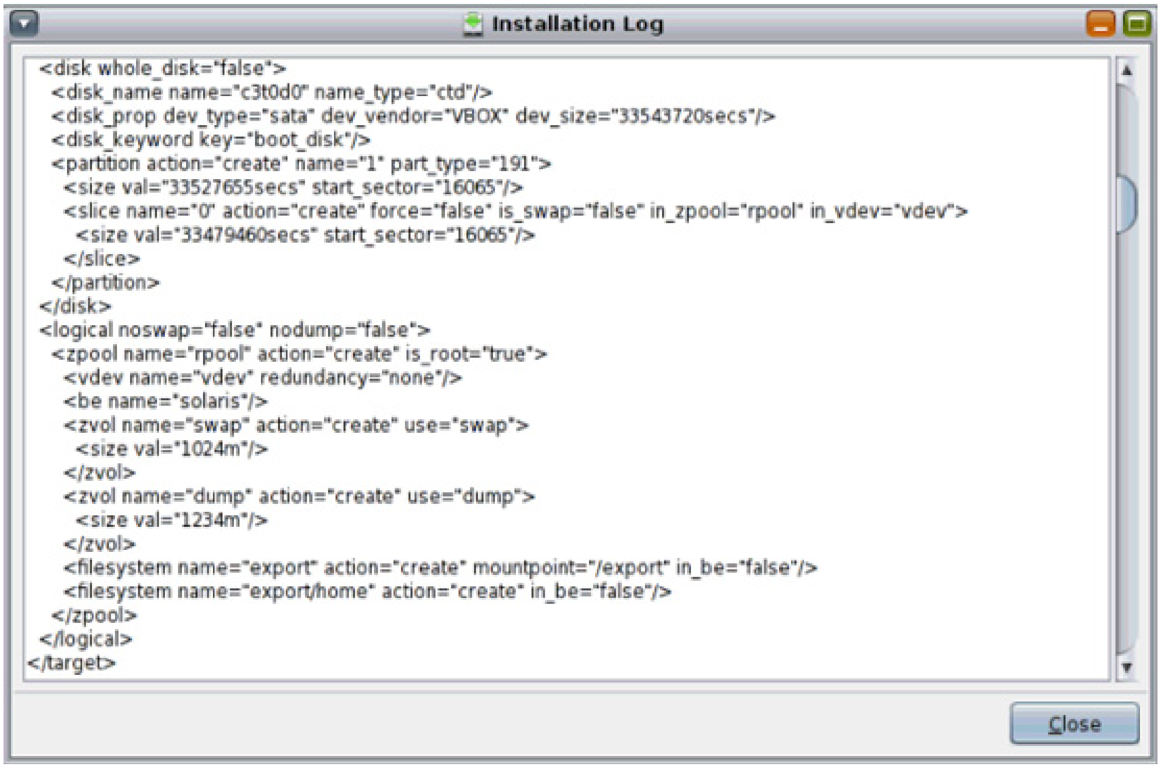
Reviewing the Installation Log



An example of the installation log is shown in this slide. The log contains a complete record of each step of the installation process. Log files are an important tool in a system administrator's tool box, so spend a few minutes acquainting yourself with the log contents.

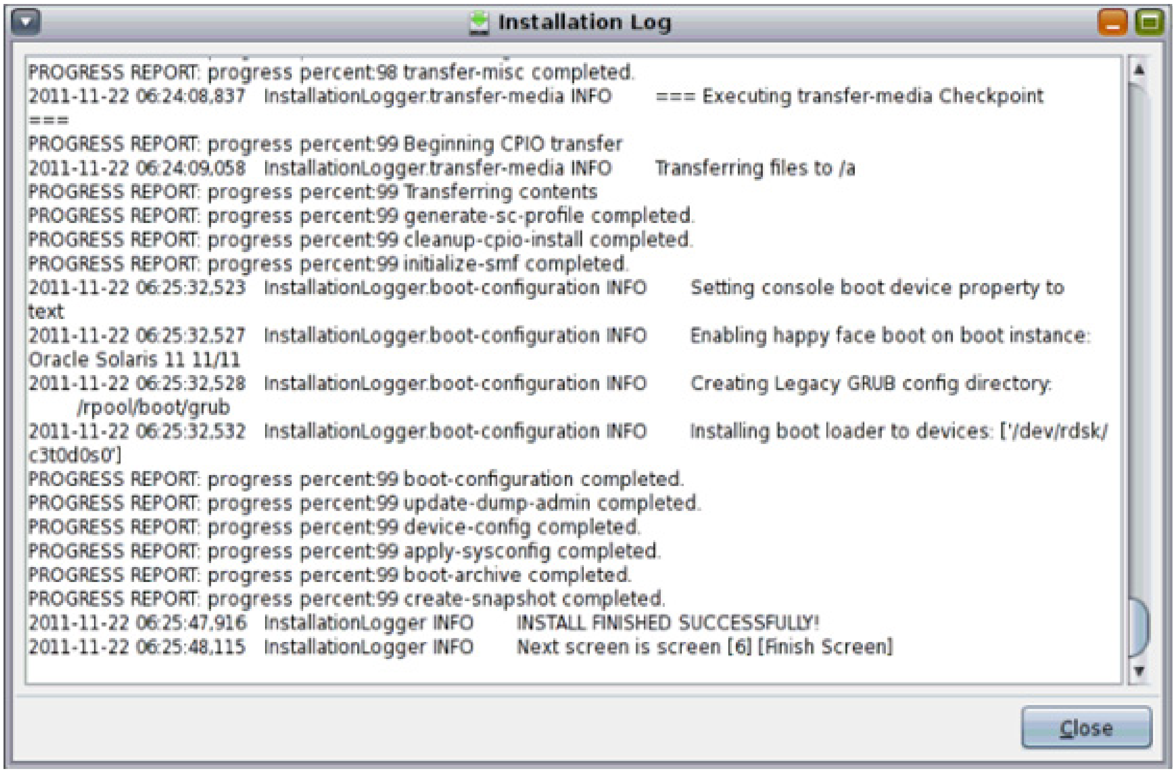
During the first part of the installation process, the configuration settings you gave during the installation process are being captured and applied to the target device.

Reviewing the Installation Log



Here, you can see that the root pool `rpool` is being created.

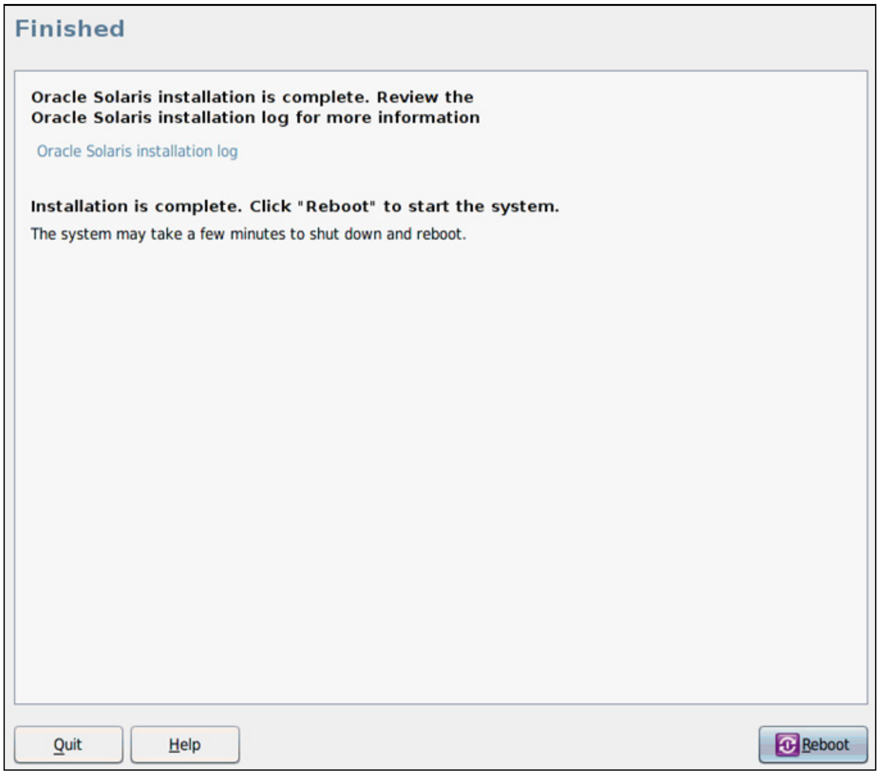
Reviewing the Installation Log



During the last stage of the installation process, boot configuration takes place. The final step is for a snapshot to be created. This snapshot captures the state of the system at this particular time. As you can see, the installation completed successfully.

When you finish reviewing the installation log and have verified that no error messages were generated, you can return to the Finish screen by clicking Close.

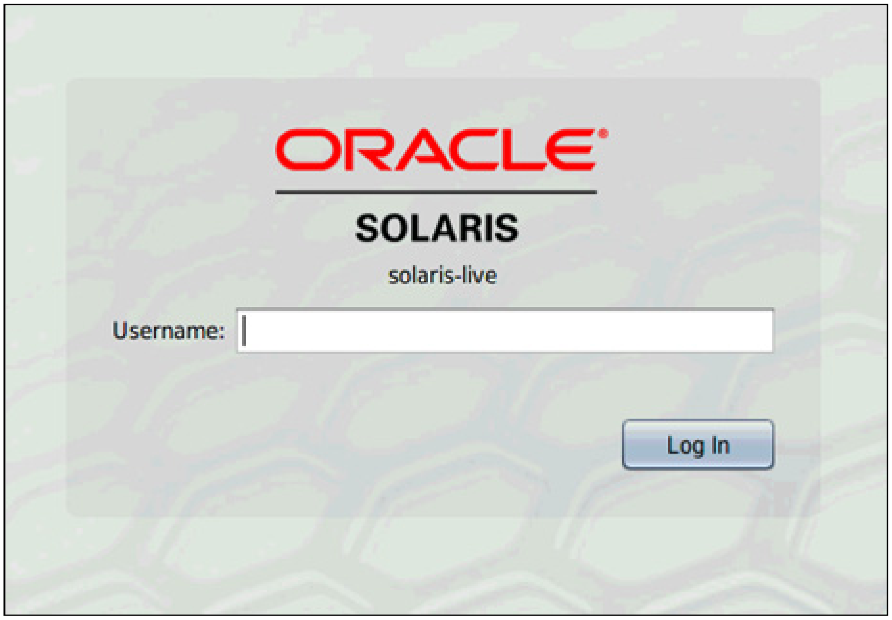
Rebooting the System



After you have verified that the installation was successful, you can reboot the system by clicking Reboot, or you can exit the installer and shut down the system.

Note: After the reboot, you can find the installation log at `/var/sadm/system/logs/install_log`.

Login Screen



After the system has rebooted, you should see the login screen. Logging in is discussed as part of the next topic.

Now you perform your own LiveCD install in Practice 2-1.

Practice 2-1 Overview: Installing Oracle Solaris 11 by Using the GUI Installer on the LiveCD

This practice covers the following topics:

- Launching the GUI
- Installing the OS
- Verifying the installation by reviewing the installation log
- Rebooting the system

In the practices for Lesson 2, you are presented with three tasks designed to reinforce the concepts presented in the lecture portion of this lesson. You will have the chance to perform the following tasks:

- **Practice 2-1:** Installing Oracle Solaris 11 by using the GUI installer on the LiveCD
- **Practice 2-2:** Installing Oracle Solaris 11 by using the text installer
- **Practice 2-3:** Verifying the operating system installation

You will find Practice 2-1 in your *Activity Guide*. It should take you about 1.5 hours to complete.

Installing Oracle Solaris 11 by Using the Text Installer

```
Welcome to the Oracle Solaris installation menu

        1   Install Oracle Solaris
        2   Install Additional Drivers
        3   Shell
        4   Terminal type (currently sun-color)
        5   Reboot

Please enter a number [1]: _
```


To begin, you are prompted to select the keyboard layout and language (just as you did for the LiveCD GUI installation), after which you see the text installer menu shown in this slide. The menu contains several options:

- **Install Oracle Solaris:** Allows you to install the operating system
- **Install Additional Drivers:** Allows you to install any device drivers that may be required to support the operating system
- **Shell:** Provides a shell; can be used to debug the installation
- **Terminal type (currently sun-color):** Provides a means of displaying the text installer correctly in case the default terminal type doesn't work
- **Reboot:** Enables you to reboot the system after the installation has completed

Initiating the Installation with Text Installer

Select option 1 to initiate the installation.

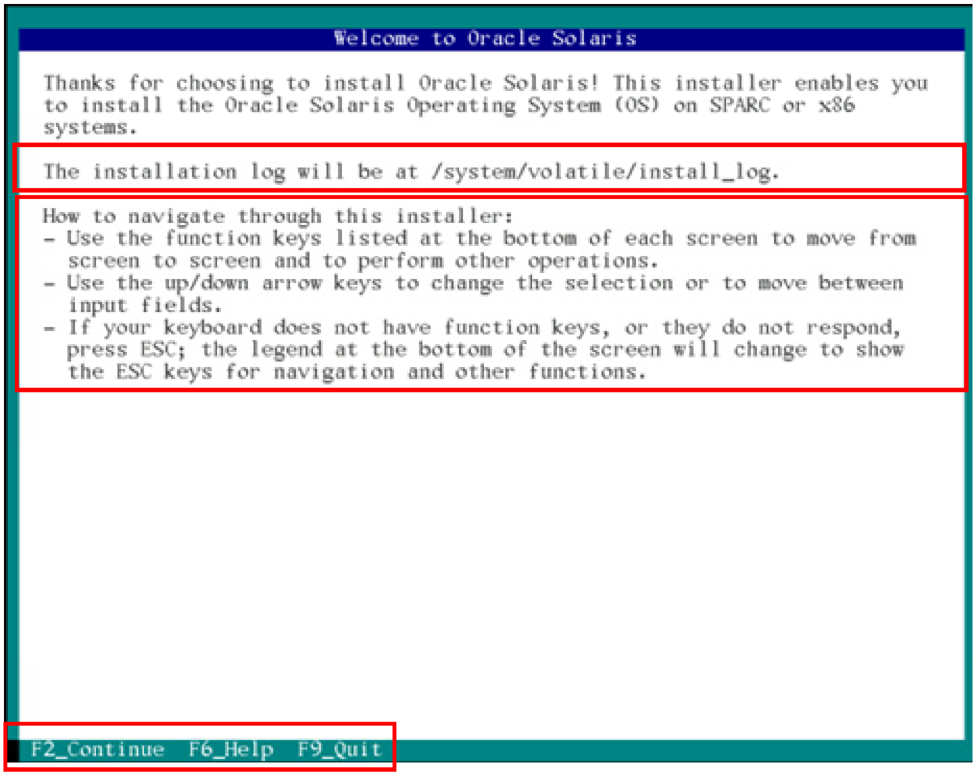
```
Welcome to the Oracle Solaris installation menu

        1  Install Oracle Solaris
        2  Install Additional Drivers
        3  Shell
        4  Terminal type (currently sun-color)
        5  Reboot

Please enter a number [1]: _
```


To initiate the installation, select option 1. The default, as indicated by the number 1 in brackets, is to install Oracle Solaris, so all you have to do is press Enter to continue.

“Welcome to Oracle Solaris” Screen



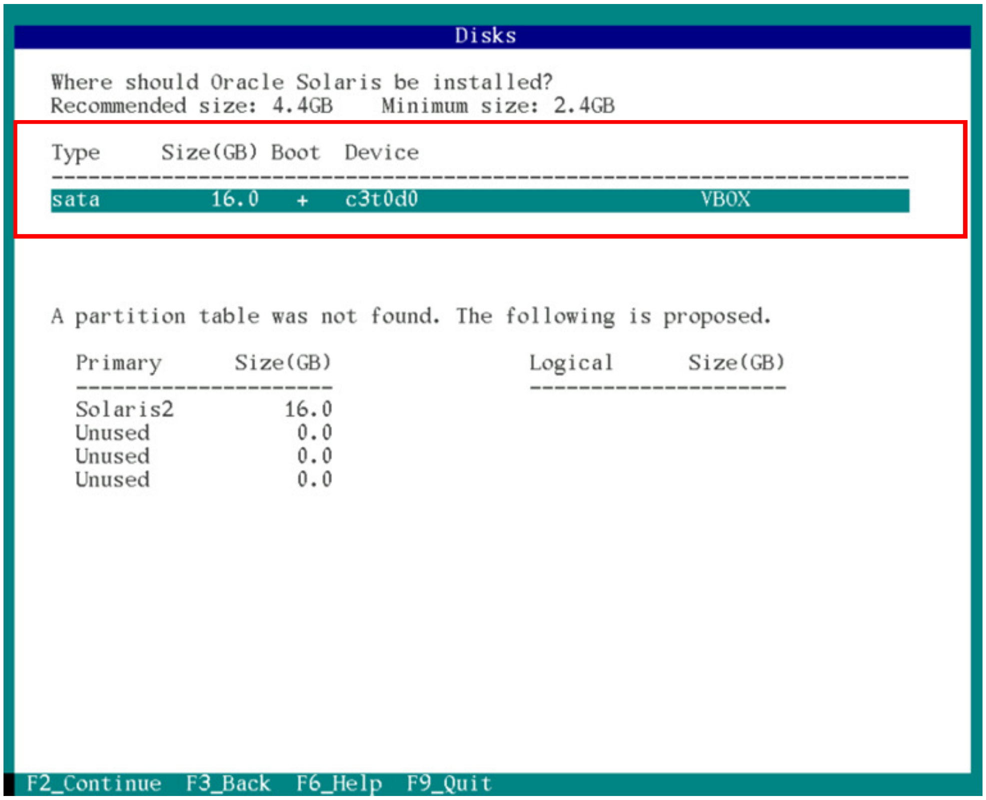
This screen provides you with the temporary location of the installation log (`/system/volatile/install_log`) as well as instructions on how to navigate through the installer by using the function keys located at the bottom of the screen and the up and down arrow keys.

You begin the installation by providing configuration data for the disk, time zone, locale, and users. The information you provide should be based on the installation plan you were given to follow.

After you have completed the configuration data screens, the actual installation begins. After the installation is complete, you are done.

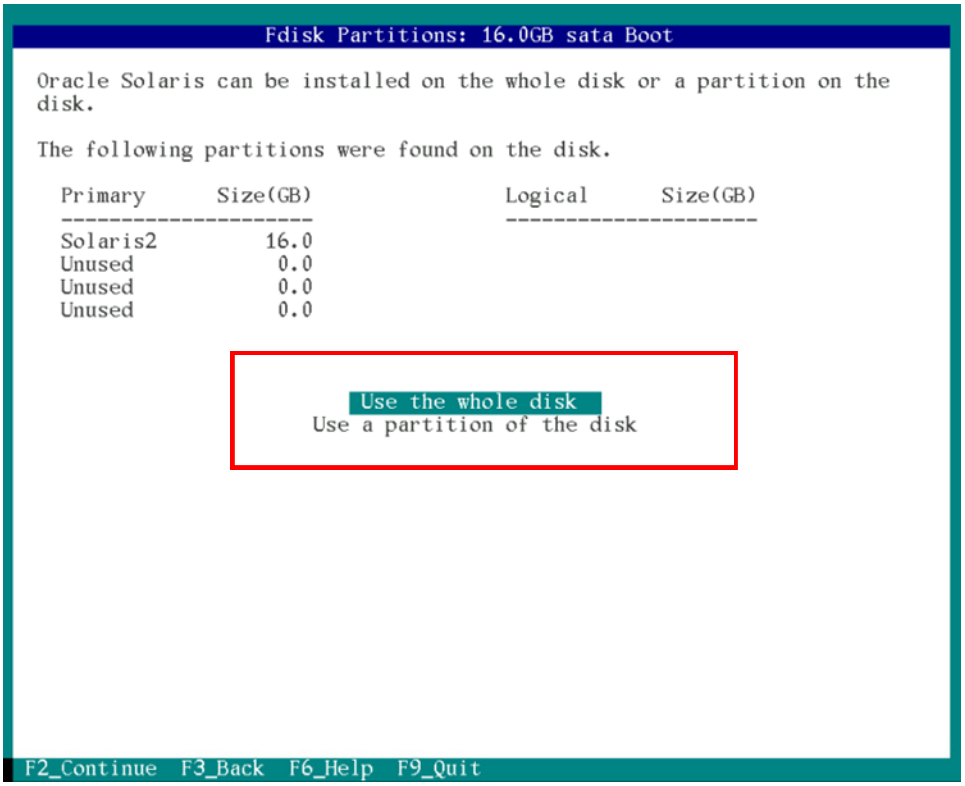
You now walk through each of the installation screens to be introduced to them. To continue to the Disk screen, press the F2 function key.

Selecting a Disk



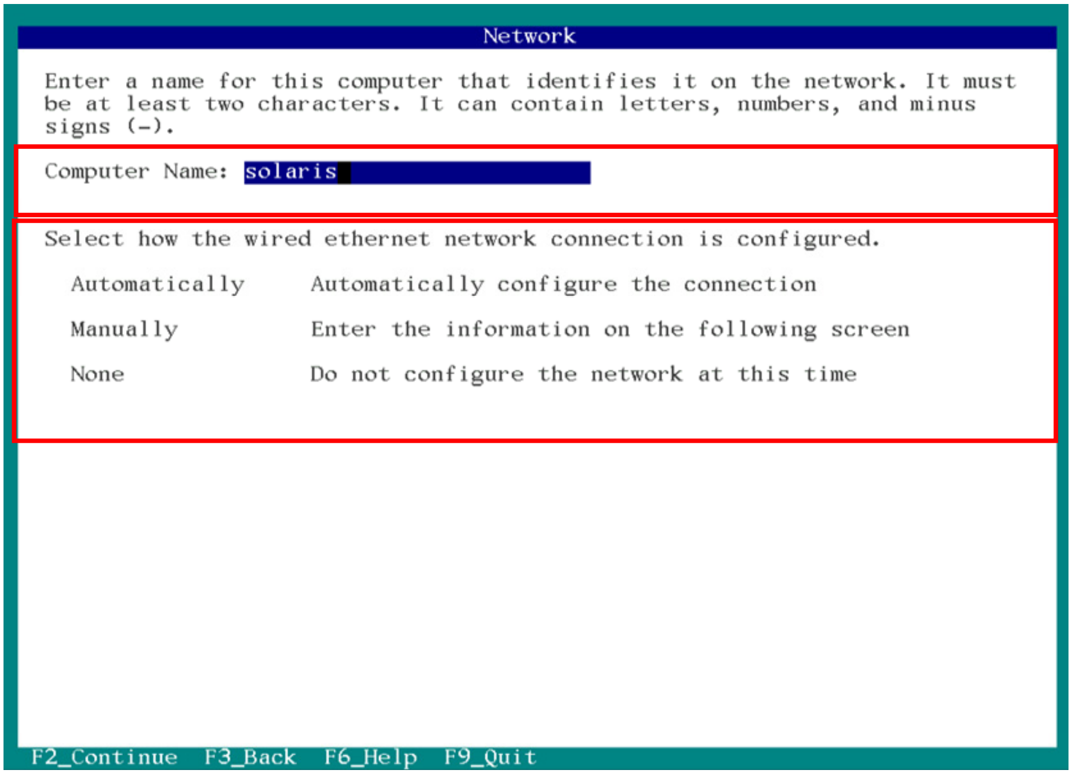
From the Welcome screen, you are taken to the Disks screen, where you are prompted to select where you want Oracle Solaris to be installed. To select the highlighted disk and continue to the next screen, press F2.

Selecting an Fdisk Partition



From the Disks screen, you are taken to the Fdisk Partitions (or the formatting disks) screen, where you can select to use a whole disk or to partition the disk. The “Use the whole disk” option is highlighted by default. Using the whole disk is highly recommended. To select this option and continue to the next screen, press F2.

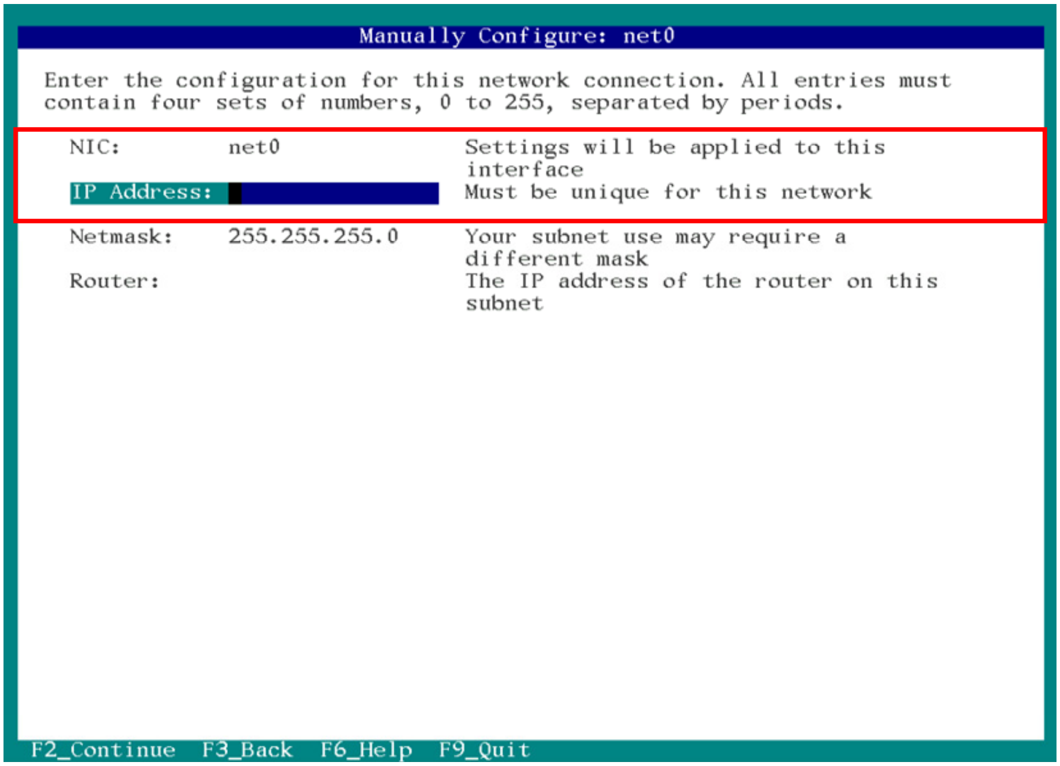
Selecting a Network



The next screen is the Network screen, where you provide a computer name and select the wired Ethernet network connection configuration (Automatically, Manually, or None). The Automatically option automatically configures the network connection for you. The Manually option enables you to manually configure the network connection by responding to the prompts presented on the subsequent screens. The None option tells the system that you do not want to configure the network at this time.

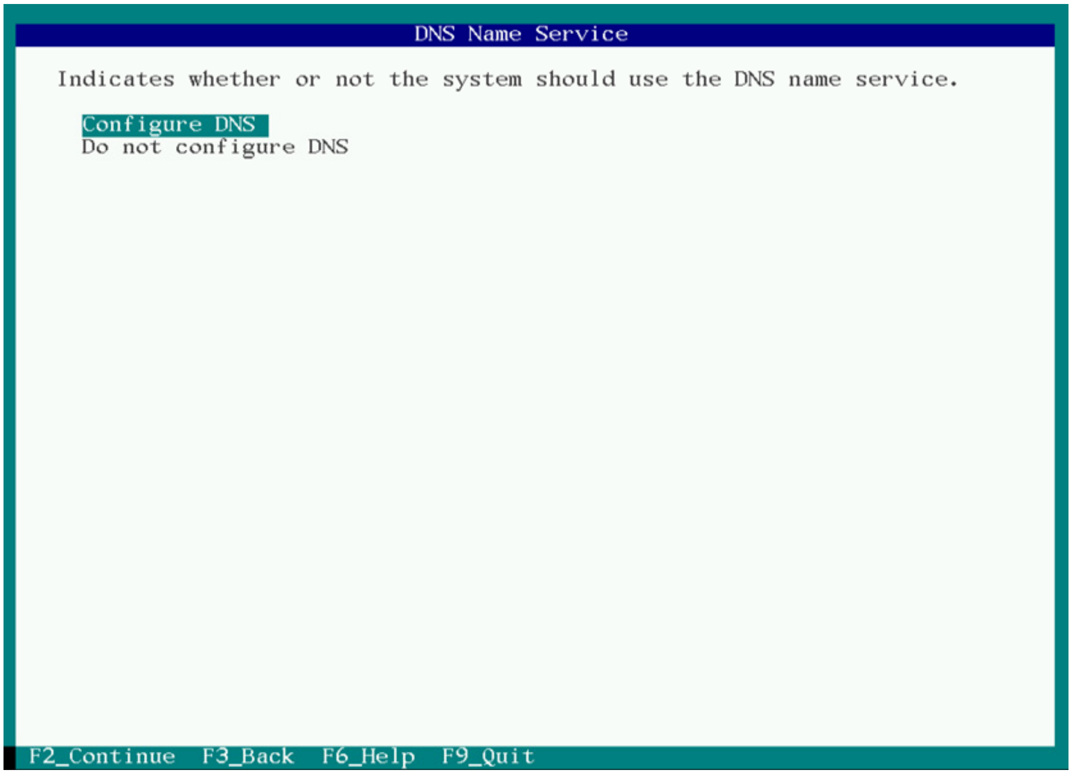
To select one of these options, arrow down to Automatically, Manually, or None. When you have the desired option highlighted, press F2 to select it and continue. For training purposes, you will manually configure the network.

Manually Configuring the Network



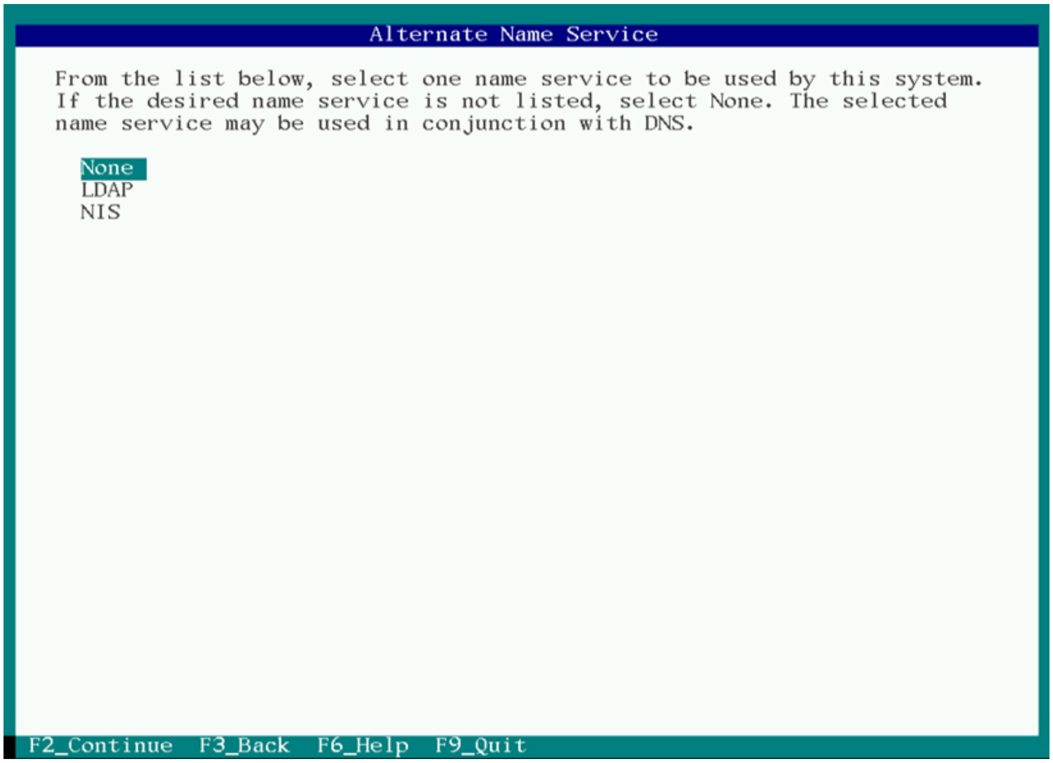
The next screen prompts you for the IP address for the network interface. In this example, the network interface is `net0`. When you have entered the required information for your network configuration, press F2 to continue.

DNS Name Service



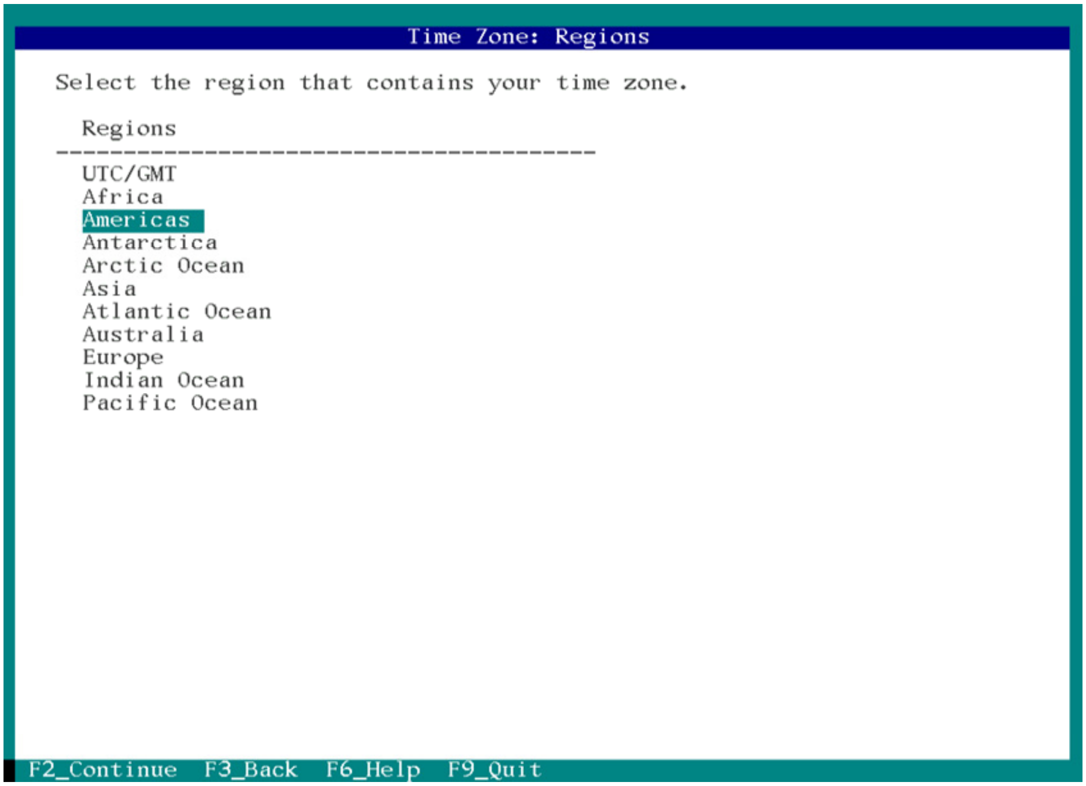
The next screen provides you the opportunity to configure a DNS name service. To configure a DNS name service, select the first option. If you do not want to configure DNS, select the second option. In this example, you are not going to configure DNS.

Alternate Name Service



The next screen enables you to select an alternate name service, such as LDAP or NIS. For this example, you are going to select None.

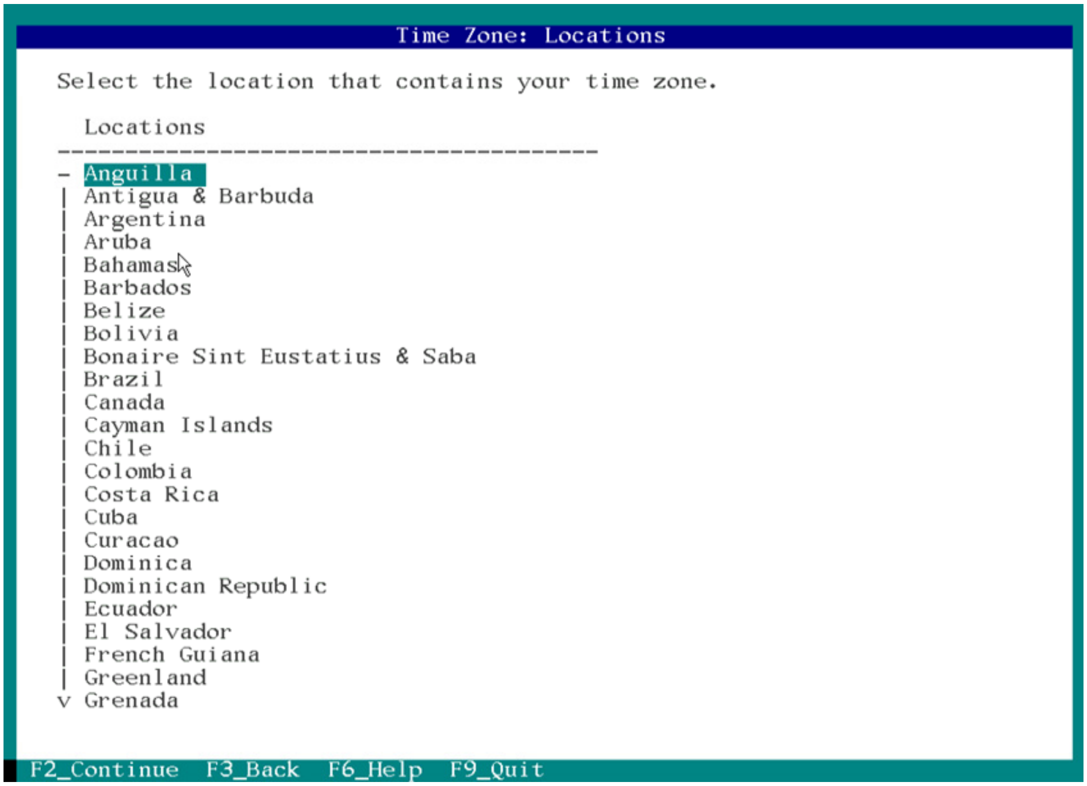
Selecting the Time Zone: Regions



The next screen that appears is Time Zone: Regions, where you select the region that contains the time zone that is appropriate to your installation. To make your selection, arrow down to the region and then press F2 to select it and continue.

In this example, you are going to select Americas.

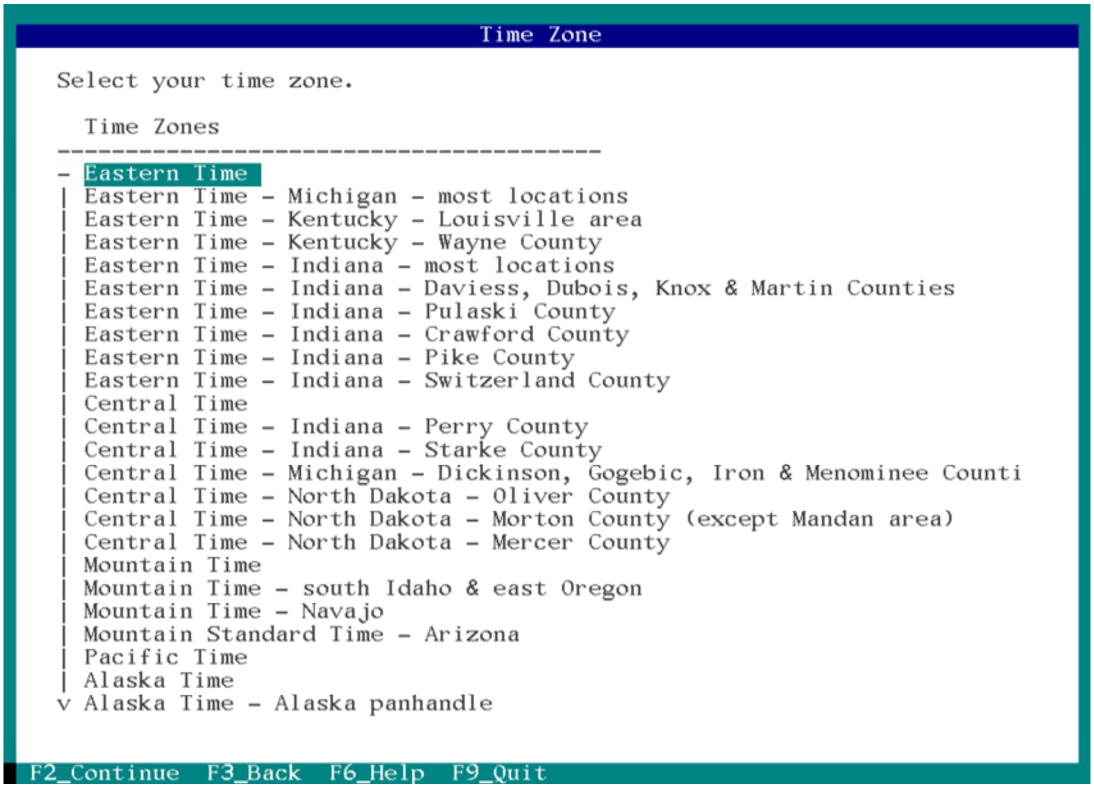
Setting the Time Zone: Locations



The next screen is the Time Zone: Locations screen, where you select the location that contains the time zone that is appropriate to your installation. Again, to make your selection, arrow down to the location of your choice and then press F2 to select it and continue.

In the sample installation, United States is selected for the location (not shown in the slide).

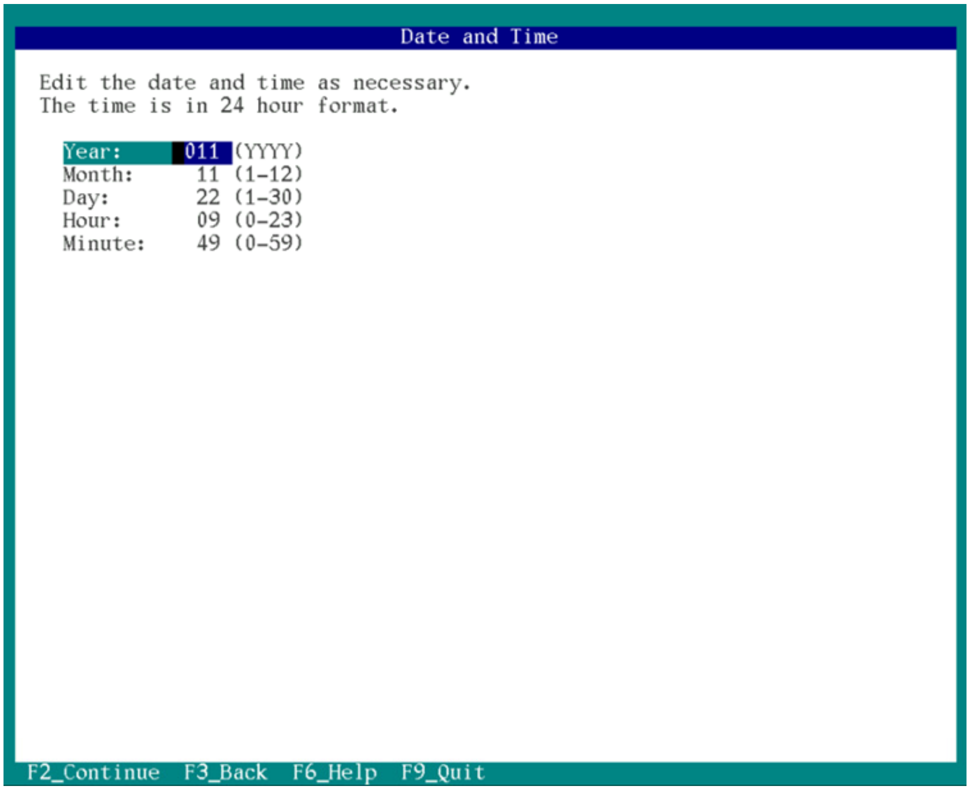
Selecting the Time Zone



You next see the Time Zone screen. The selections presented on this screen are based on the region and location selections you made previously. To select the appropriate time zone, arrow down until your time zone is highlighted, and then press F2.

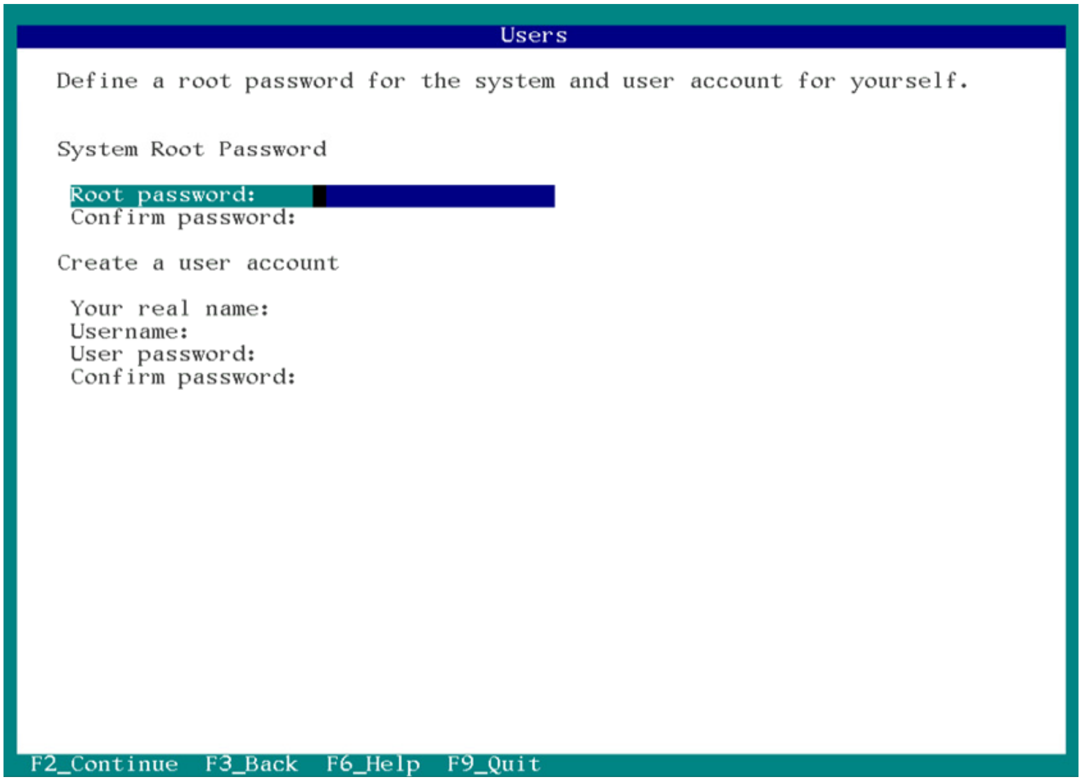
To support the example installation, you select Mountain Time.

Setting the Date and Time



The “Date and Time” screen appears next. Review the information that is presented and edit it as required. Note that the time is in the 24-hour format. After you have made the necessary edits, press F2 to continue.

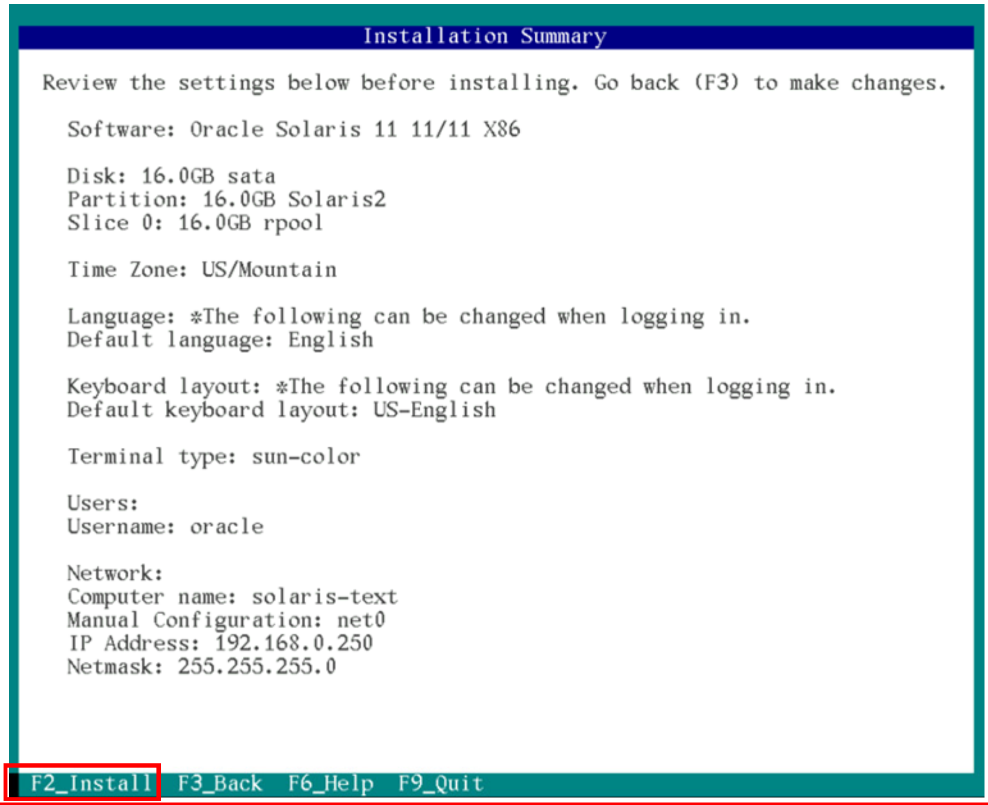
Providing User Information



Next is the Users screen, where you enter your user information to include the system root password, your name, your username, and your user password. To continue, press F2.

Note: If you provide a username, that user is given the `root` role. If you do not provide a username, `root` is an account rather than a role, and is set to expire immediately.

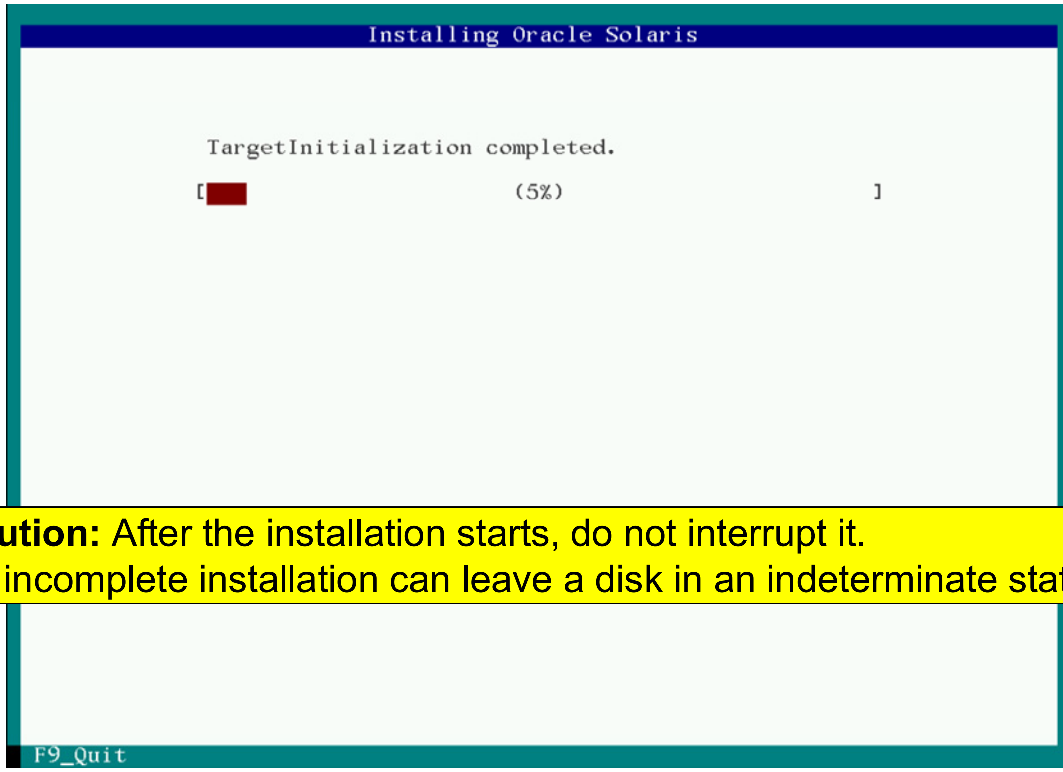
Reviewing Installation Summary



After completing the configuration data, you see the Installation Summary screen. Review the information carefully to make sure it is accurate before you start the installation. To start the installation, press F2.

Note: You can go back and make changes if you need to by pressing F3.

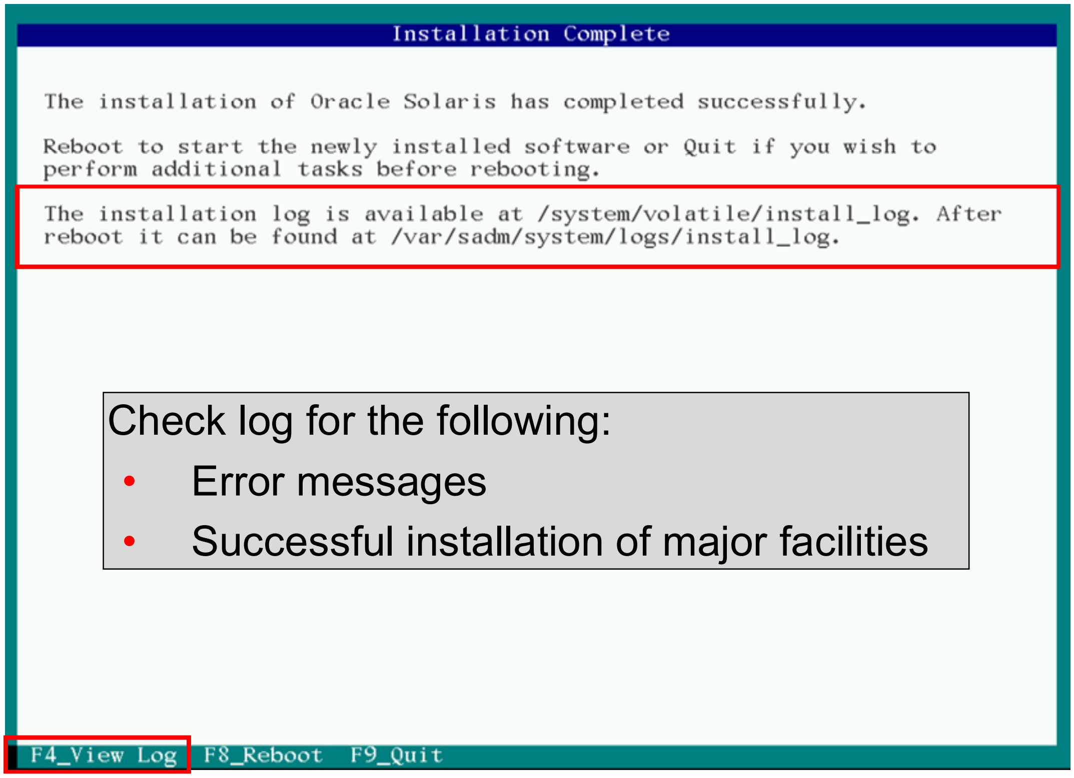
Monitoring the Installation



Caution: After the installation starts, do not interrupt it.
An incomplete installation can leave a disk in an indeterminate state.

The Installing Oracle Solaris screen enables you to monitor the progress of the installation. The installation takes about 10 to 15 minutes to complete.

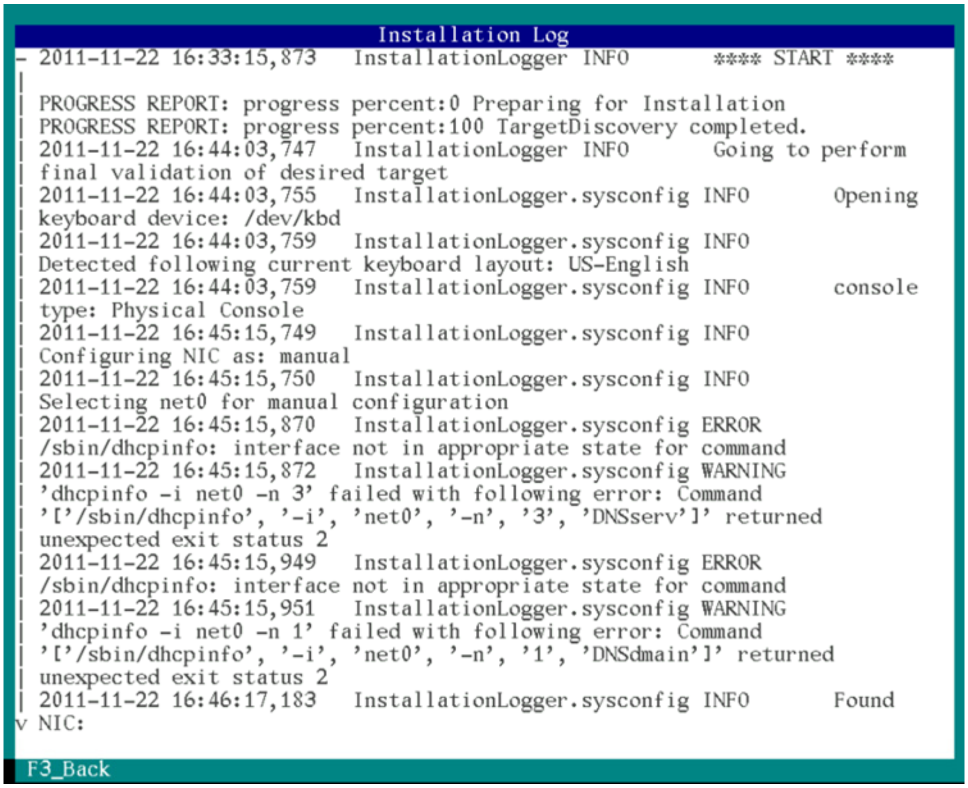
Verifying the Installation



When the installation concludes, the Installation Complete screen appears. This screen provides you with access to the installation log and, as you saw with the LiveCD installation, an opportunity to verify that no error messages were generated and that all the major facilities installed successfully.

You have the option of viewing the log at its `/system/volatile/install_log` location before rebooting by pressing F4. Alternatively, you can view the log at `/var/sadm/system/logs/install_log` after reboot.

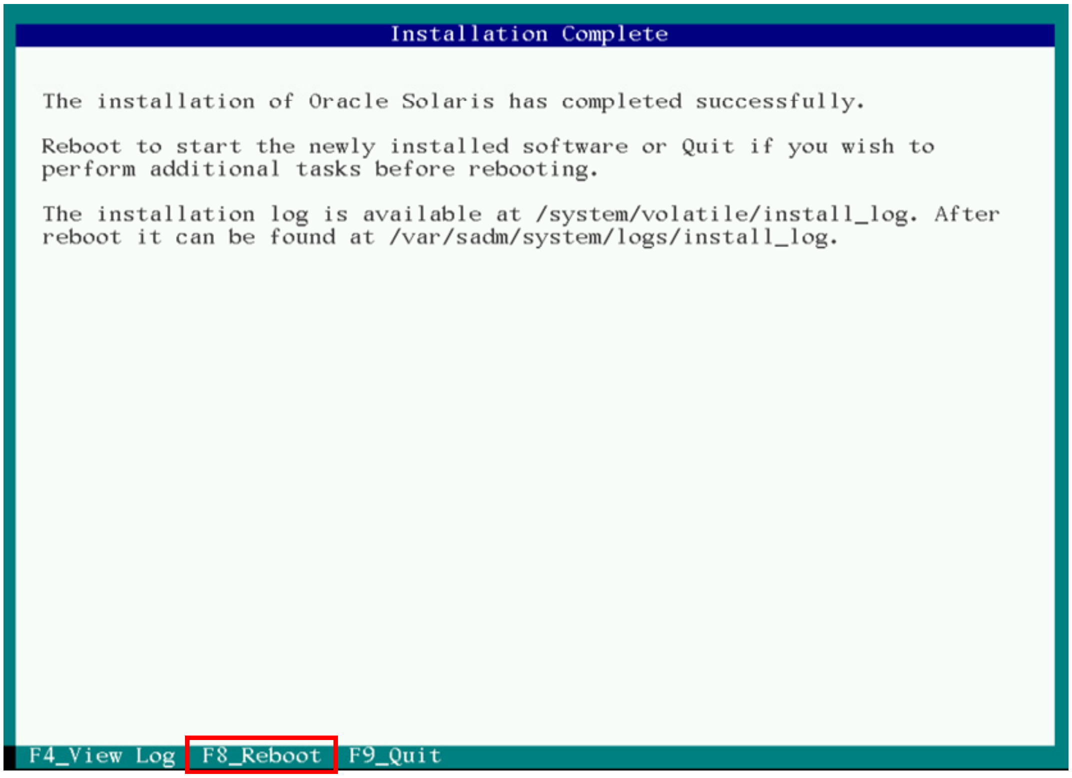
Reviewing the Installation Log



An example of the installation log for the text install is shown in this slide. As with the LiveCD installation log, the text install log contains a complete record of each step of the installation process. You should always take time to review the log carefully to make sure the operating system has been installed as planned.

When you are done reviewing the log, press F3 to return to the Installation Complete page.

Rebooting the System



After you have verified that the installation was successful, you can reboot the system by pressing F8.

If you press F9 to quit, you are returned to the text installer menu, where you can select option 5 to reboot the system.

Login Screen

```
SunOS Release 5.11 Version 11.0 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Loading smf(5) service descriptions: 170/170
Hostname: solaris-text
Configuring devices.
Loading smf(5) service descriptions: 8/8

Solaris-text console login: _
```


After the system has rebooted, you should see the login screen. Logging is discussed as part of the next topic.

Now you perform your own text install in Practice 2-2.

Practice 2-2 Overview: Installing Oracle Solaris 11 Using the Text Installer

This practice covers the following topics:

- Launching the installer
- Manually configuring the network
- Installing the OS
- Verifying the installation
- Rebooting the system

This practice should take you about one hour to complete.

Lesson Agenda

- Planning for an Oracle Solaris 11 OS Installation
- Installing Oracle Solaris 11 by Using an Interactive Installer
- Verifying the Operating System Installation

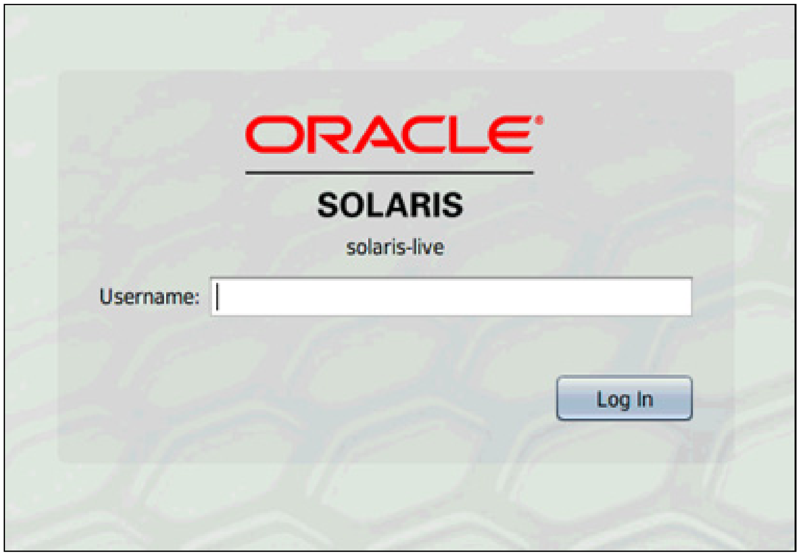
In this section, you learn about the importance of verifying your newly installed operating system.

Verifying the Operating System Installation

- Verifying login information
- Verifying the system's host name
- Displaying basic system information
- Displaying a system's release information
- Displaying disk configuration information
- Displaying the installed memory size
- Displaying information about network services
- Displaying network interface information

After you have installed Oracle Solaris 11 on your system, you want to verify the new operating system installation as well as document key information about the system that can be used as a baseline for change management documentation. You begin by verifying that the user login is functioning correctly.

Checking the Login Username




```
solaris-text console login:
```

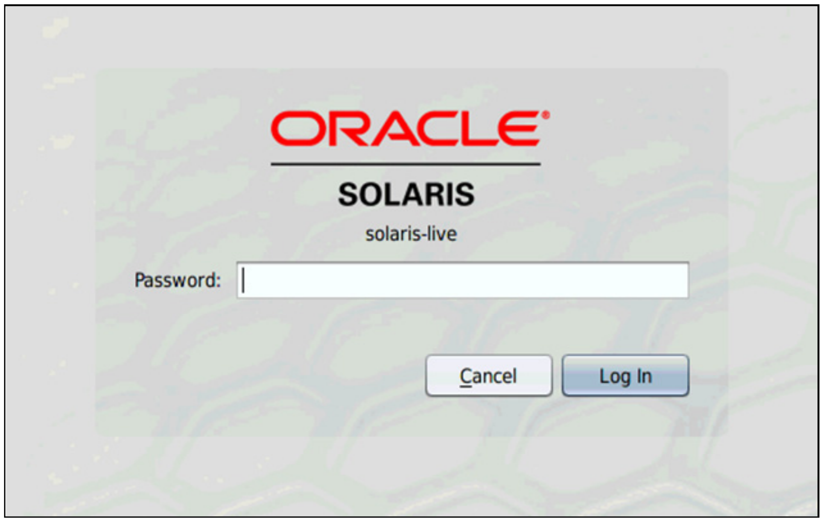

LiveCD GUI Environment (`solaris-live` system)

At the login screen (as shown at the top of the slide), enter the username you established during installation and then click Log In. If the username is correct, the password screen appears.

Text Install Environment (`solaris-text` system)

At the console login (as shown in the code example), enter the username you established during installation and then press Enter. If the username is correct, the password prompt appears.

Checking the Login Password




```
Password:
Oracle Corporation  SunOS 5.11  11.0  November 2011
oracle@solaris-text:~$
```

ORACLE

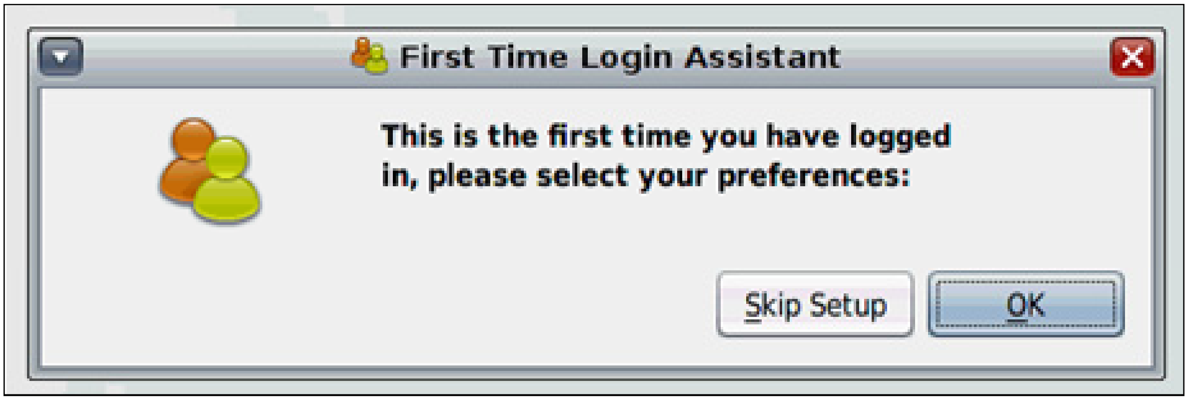
LiveCD GUI Environment (`solaris-live` system)

To check the login password, enter the password you provided during the installation and click Log In (in the password screen, as shown at the top of the slide). If the password is correct and this is the first time you are logging in, you then see something called the First Time Login Assistant. You will take a closer look at this tool shortly.

Text Install Environment (`solaris-text` system)

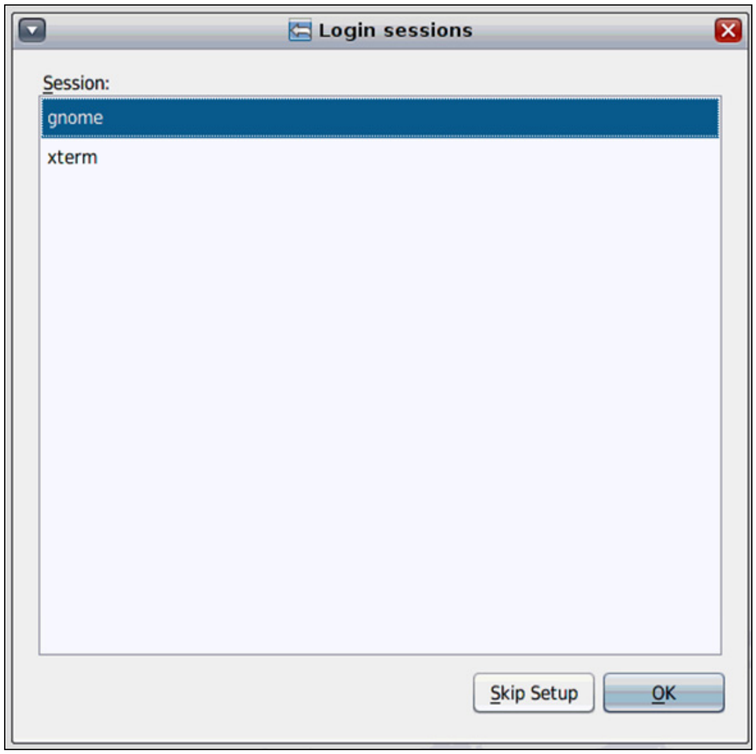
To check the login password, go to the password prompt and enter the password you provided during the installation and press Enter. If the password is correct, you are presented with the operating system release information and the command-line user prompt (as shown in the code example in the slide).

Using the First Time Login Assistant (LiveCD GUI)



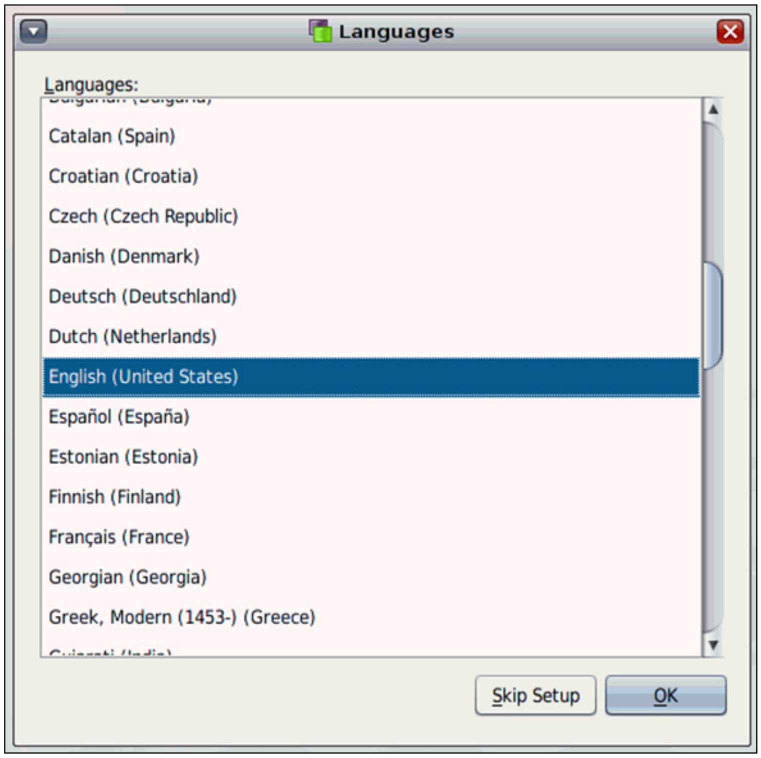
If you are in the LiveCD GUI environment, you will be presented with the First Time Login Assistant the first time you log in. The First Time Login Assistant enables you to select a login session type and language. If you think you don't need this type of assistance, you can click Skip Setup. If you click OK, the Login Session window appears.

Selecting a Login Session (LiveCD GUI)



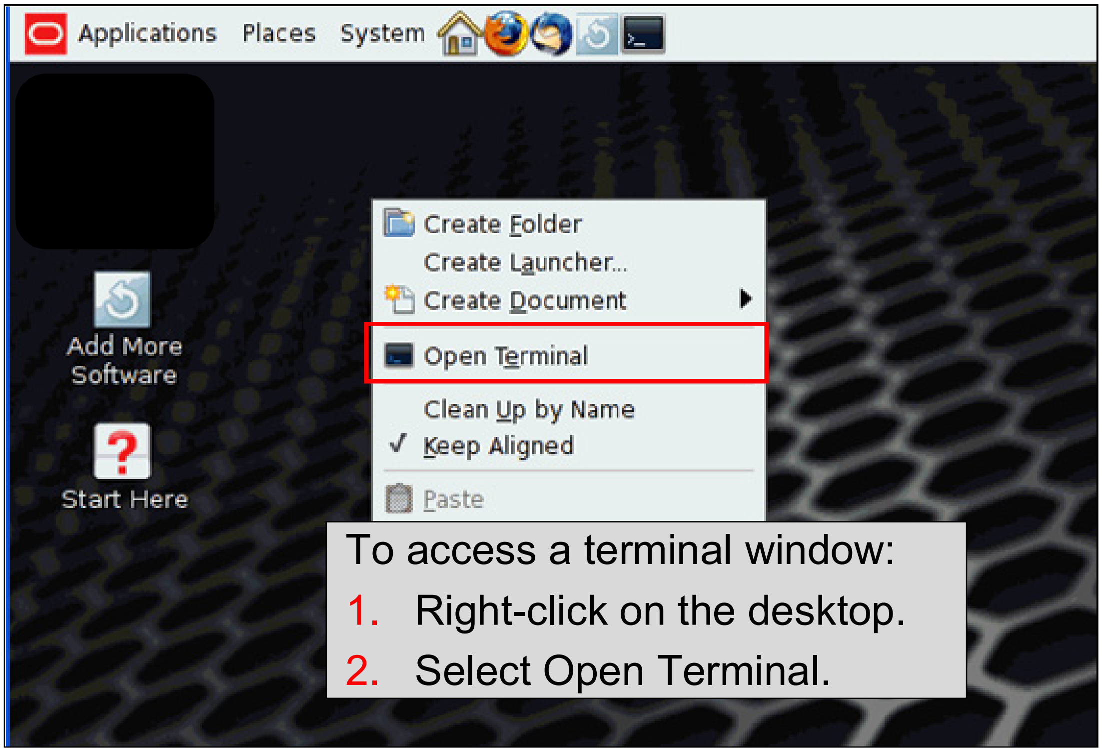
The "Login sessions" window allows you to select the type of login session you want to use. Your choices are gnome or xterm. Gnome is a desktop session, and xterm is a terminal window. To select a session, highlight it and click OK. We are going to select gnome.

Selecting a Language (LiveCD GUI)



The next window is the Languages window. To select a language, highlight it and click OK.

Accessing a Terminal Window from Gnome (LiveCD GUI)



If you selected the Gnome session, the desktop environment appears. To work with the operating system by using the command line, you need to open a terminal window. To do so, right-click on the desktop to open the menu and then select Open Terminal.

When you start working in the terminal window, the commands you use to communicate with the operating system are the same, regardless of whether you are in the LiveCD environment or the text install environment.

Verifying the Host Name

To display the host name, use `hostname`.

```
$ hostname
solaris-live
```

The host name should match the computer name that you provided during installation.

When you are in the terminal window, the first thing you want to verify is the host or node name that appears in the command prompt. The name that appears here should match the computer name that you provided during installation. The host name is the name by which the system is known to a communications network. In this example, the host name is `solaris-live`.

Note: If you were performing the verification on the text install machine, your host name would appear as `solaris-text` (if that is the name you gave the system during the installation).

You can also verify the host name by using the `hostname` command.

Displaying Basic System Information

To display basic information about the system, run `uname -a`.

```
$ uname -a
SunOS solaris-live 5.11 11.0 i86pc i386 i86pc
```

This system's basic information:

- Operating system: SunOS
- Hostname: solaris-live
- Release: 5.11
- Version: 11.0
- Hardware name: i86pc i386
- Processor type: i386

When you have verified the host name, the next thing you want to do is display basic information about the system.

You use the `uname -a` command to display the information listed in the slide. The operating system release level is, for example, Oracle Solaris 10 or Oracle Solaris 11.

Capturing this information about your system for baseline purposes is extremely important for updating software packages, which is discussed in the next lesson.

Note: You can also run the `uname` command with specific options to display any one of the information items individually. See the `uname`(1) man page for details.

Displaying a System's Release Information

To display the operating system's release information, run `cat /etc/release`.

```
$ cat /etc/release
                      Oracle Solaris 11 11/11 X86
    Copyright (c) 1983, 2011, Oracle and/or its affiliates.
                           All rights reserved.
                        Assembled 18 October 2011
```


The Oracle Solaris 11 operating system has a file called `/etc/release` that contains information about the system: the full operating system name, version of the release, hardware architecture, copyright, and the date on which the release was assembled. You use `cat /etc/release` to display the contents of this file.

Note: Use the `cat /etc/release` command (instead of `uname -a`) to get current update release information, such as release 11 versus 10.

Displaying Disk Configuration Information

To display disk information, switch to superuser and run `format`.

```
$ su -
Password:
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c7t0d0 <SUN-DiskImage-40GB cyl 2085 alt 2 hd 255 sec 63>
          /pci@0,0/pci-ide01,1/ide@0/cmdk@0,0
Specify disk (enter its number):0
No Alt slice
No defect list found
[disk formatted, no defect list found]
/dev/dsk/c7d0s0 is part of active ZFS pool rpool1. Please see zpool
     (1M).

<continued on next page>
```

Note: The format utility requires superuser privileges.

You use the `format` command to display information about the disks that are on the system. Recall that we used the entire disk for the installation and that the disk name was `c7t0d0`.

`format` is a disk partitioning and maintenance utility. It allows you to format, label, repair, analyze, and scrub data off a disk. To execute the `format` command, you must have root privileges. To switch to root, you can use the command `su -`.

Note: `su` stands for "switch user."

In the example in the slide, only one disk has been configured in the system: `c7t0d0`. To see how a disk is formatted, you enter the number of the disk (in this example, `0`) after the `Specify disk (enter its number)` prompt, and then press Enter.

Displaying Disk Configuration Information: Format Menu

To display disk partition information, select `verify`.

```
FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        fdisk      - run the fdisk program
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format> verify
<continued on next page>
```


After you have specified a disk, the Format Menu appears (as shown in the slide). To see how the disk has been partitioned (that is, how the slices on the disk have been allocated), enter `verify` at the `format` prompt and then press Enter.

Displaying Disk Configuration Information: Partition Table

```
Primary label contents:

Volume name = <        >
ascii name  = <DEFAULT cyl 2085 alt 2 hd 255 sec 63>
pcyl        = 2087
ncyl        = 2085
acyl        =    2
bcyl        =    0
nhead       =  255
nsect       =   63
Part      Tag    Flag     Cylinders         Size            Blocks
  0        root    wm       1 - 2084        15.96GB     (2084/0/0) 33479460
  1 unassigned    wm       0               0           (0/0/0)           0
  2      backup    wu       0 - 2084        15.97GB     (2085/0/0) 33495525
  3 unassigned    wm       0               0           (0/0/0)           0
  4 unassigned    wm       0               0           (0/0/0)           0
  5 unassigned    wm       0               0           (0/0/0)           0
  6 unassigned    wm       0               0           (0/0/0)           0
  7 unassigned    wm       0               0           (0/0/0)           0
  8        boot    wu       0 -     0        7.84MB     (1/0/0)       16065
  9 unassigned    wm       0               0           (0/0/0)           0
format>quit
#
```


The partition table shows partition by partition (or slice by slice) how the space on a disk is being used. The disk shown in this example (the disk on which we selected to install the operating system) has nine partitions, with the first partition (partition `0`) being allocated to contain the root file system. Partitions `1`, `3`, `4`, `5`, `6`, `7`, and `9` are currently unassigned. Partition `2` represents the whole disk, and partition `8` contains the boot program and is used in the booting process.

You will examine in more detail how to partition a disk and how to interpret the partition table in the lesson titled “Setting Up and Administering Data Storage.” For now, it is enough that you know where to go to view the disk configuration and how to get there by using the `format` utility.

When you are done displaying the current disk configuration, you can leave the `format` utility and return to the root user prompt by entering `quit` at the `format` prompt.

Displaying Installed Memory Size

To display memory size, use `prtconf | grep Memory`.

```
# prtconf | grep Memory
Memory size: 1024 Megabytes
```


Another piece of information that you want to capture for your baseline documentation is the total amount of physical memory installed on the system. Oracle Solaris provides the `prtconf` command to print the configuration that was initially created by the lower-level hardware discovery phase. This configuration includes the quantity of memory.

To display only the memory information, use `prtconf | grep Memory` (as shown in the example in the slide).

Memory plays an important role in the performance of a system, so it is important to monitor how much memory the system is using. In this example, we have 1024 MB of installed memory.

Displaying Information About Network Services

To display information about network connection configuration services, run `svcs network/physical`.

```
# svcs network/physical
disabled  15:35:29 svc:/network/physical:nwam
online    15:35:09 svc:/network/physical:upgrade
online    15:35:09 svc:/network/physical:default
```


The current network setup is also important information to capture for your baseline documentation. Oracle Solaris has a feature called the Service Management Facility (SMF) that controls the operating system services. To see these services, use the `svcs` command.

Note: We discuss SMF in the lesson titled “Administering Services.”

To display the network services information, use the `svcs network/physical` command. In this example, you can see that the `network/physical:default` service is online.

Displaying Network Interface Information

To display network interface information, run `ipadm show-addr`.

```
# ipadm show-addr
ADDROBJ           TYPE              STATE    ADDR
lo0/v4            static            ok       127.0.0.1/8
net0/_b           dhcp              ok       10.0.2.15/24
lo0/v6            static            ok       ::1/128
net0/_a           addrconf          ok       fe80::a00:27ff:fe4c:d1cb/10
```


Another piece of network-related information you want to capture for your baseline documentation is network interface information. You want to make a note of the network interface information on your system (for example, `net0/_b` and `net0/_a`) and verify that these network interfaces are up and running. In the lesson titled "Administering a Physical Network," you learn how to monitor the status of these network interfaces (NICs). For now, it is enough that you make a note of the NIC names and their current status.

To display the network interface information, you use the `ipadm show-addr` command. For this installation, using the LiveCD GUI install option, the IP address was obtained automatically from DHCP by NWAM.

The `ipadm` command is the tool for all IP interface configuration administration tasks. You will take a closer look at this command in the network administration lesson.

Baseline System Information Commands: Summary

System Information	Command
Host name	`hostname`
Basic information: Operating system name, release, version, host name, hardware architecture, and processor type	`uname -a`
Operating system release information	`cat /etc/release`
Disk configuration	`format`
Installed memory	`prtconf | grep Memory`
Information about network services	`svcs network/physical`
Network interface information	`ipadm show-addr`

Your verification of the operating system installation is complete. You have collected all the information you need to document the system's baseline configuration.

The table in this slide summarizes the commands used to display system information that can be used to document a baseline for your system. You can find out more about each of these commands in the man pages.

Practice 2-3 Overview: Verifying the Operating System Installation

This practice covers the following topics:

- Verifying login information
- Verifying the system's host name
- Displaying:
  - Basic system information
  - System release information
  - Boot disk configuration
  - Installed memory size
  - Network information

This practice should take you about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for an Oracle Solaris 11 operating system installation
- Install the Oracle Solaris 11 operating system by using an interactive installer
- Verify the operating system installation

In this lesson, you were introduced to the interactive and automated installation options. You learned how to perform an interactive installation by using both the LiveCD GUI installer and the text installer. You then learned how to verify the operating system installation.

3

Updating and Managing Software Packages

ORACLE

Objectives

After completing this lesson, you should be able to:

- Implement a plan for an Oracle Solaris 11 operating system software update
- Update the Oracle Solaris 11 operating system by using IPS
- Manage software packages by using the Oracle Solaris 11 operating system
- Administer boot environments

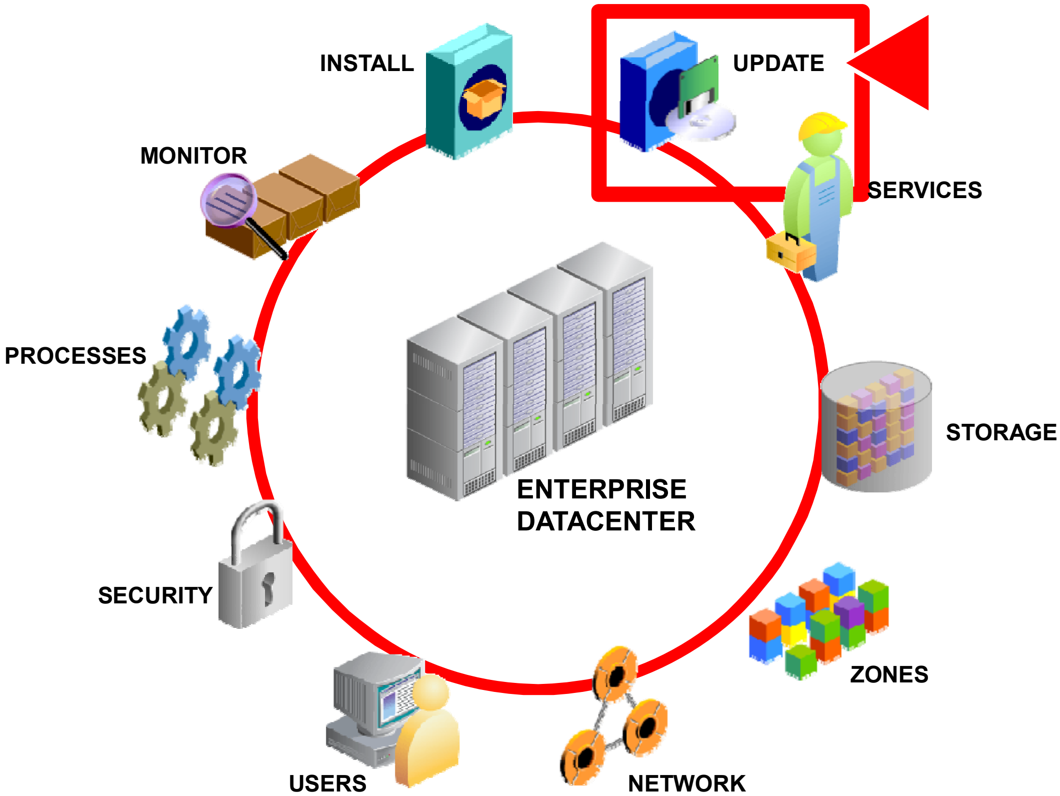
In this lesson, you learn how to update the operating system to make sure your system is running the most current release of the software. You also learn how to manage the software packages installed on the system.

The lesson begins by discussing the importance of planning for the operating system software update. This discussion includes an introduction to the Image Packaging System (IPS).

Next, it focuses on how to update the operating system using IPS, followed by how to manage software packages in Oracle Solaris 11.

The lesson concludes by introducing you to boot environments and discussing how to administer them.

Workflow Orientation



Before you begin, take a moment to orient yourself to where you are in the job workflow. You have successfully installed the operating system and captured a baseline of the system's current state.

Your next task as a system administrator is to make sure that the operating system is running the most current release of the software. That is, you want to make sure that the latest features, feature updates, and bug fixes are installed on the system. This lesson focuses on when and how you perform these updates.

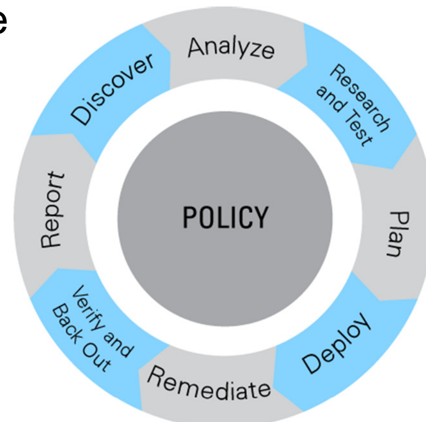
Lesson Agenda

- Planning for an Oracle Solaris 11 OS Software Update
- Updating the Oracle Solaris 11 OS by Using IPS
- Managing Software Packages
- Administering Boot Environments

Planning for an Oracle Solaris 11 OS Software Update

Planning is required to make sure the operating system is:

- Running the most current version of the software
- Updated regularly to make sure that the latest feature updates and bug fixes have been installed
- Updated in accordance with business policy, software update strategy, and plan



Oracle issues software updates on a regular and frequent basis. This means that your newly installed system is no longer current. A best practice is to update the newly installed operating system with the latest software release before you perform any more configuration activities, and then update the system on a regular basis thereafter. The frequency with which you update the operating system is based on your company's business policy. Your company's software update strategy and plan are based on the business policy.

An example of a software update strategy based on business policy is shown in the slide diagram. The strategy supports a workflow process in which the software updates are continually analyzed, tested, monitored, and evaluated for applicability to the business drivers.

Software Update Plan

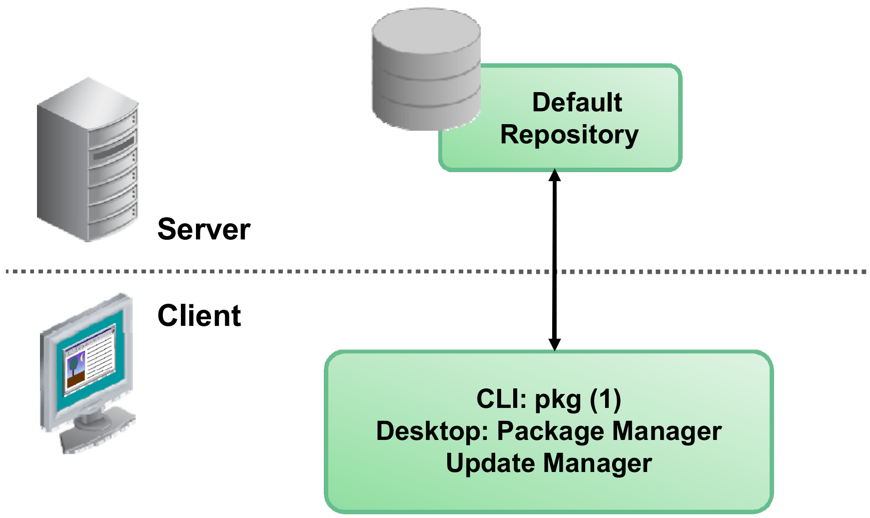
The software update plan should tell you:

- Who will perform the update
- When the update should be performed
- What systems should be updated
- What software packages should be updated
- How the update should be performed

The software update plan should contain information about who is responsible for performing the update, when the update should be performed (to have the least impact on production), what systems should be updated, what software packages should be updated, and how the update should be performed.

You now learn about the last item: how the update should be performed.

Software Update Process: Overview

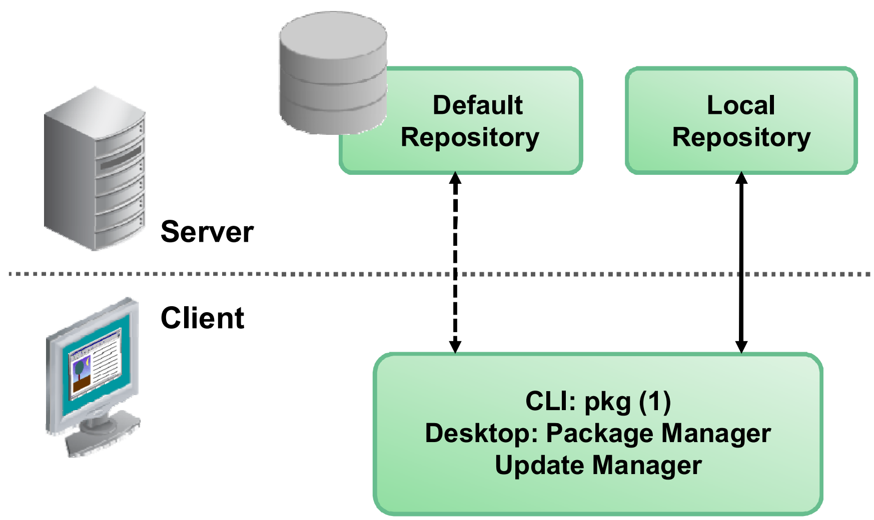


Oracle IPS default repository:
http://pkg.oracle.com/solaris/release/

Oracle uses a technology called the Image Packaging System (IPS) to manage software updates. As the Oracle Solaris 11 product engineering group releases software updates for the operating system, the updates are published as packages to a web-based Oracle repository for distribution. The repository is a collection of IPS packages. Oracle's repository is known as the *default repository* and is located at http://pkg.oracle.com/solaris/release/. You will also hear the location of the repository referred to as its *origin*.

Note: An IPS package is a collection of directories, files, links, drivers, dependencies, groups, users, and license information in a defined format.

Software Update Process: Overview



Default repository: http://pkg.oracle.com/solaris/release/
Local repository: Configured on your local network

IPS supports the ability to create local package repositories.

Note: The word *local* in this context means being able to set up a package repository that is, for example, local to your machine, your network, or your geographic area.

Having a local package repository is necessary when your network clients do not have access to Oracle's default repository.

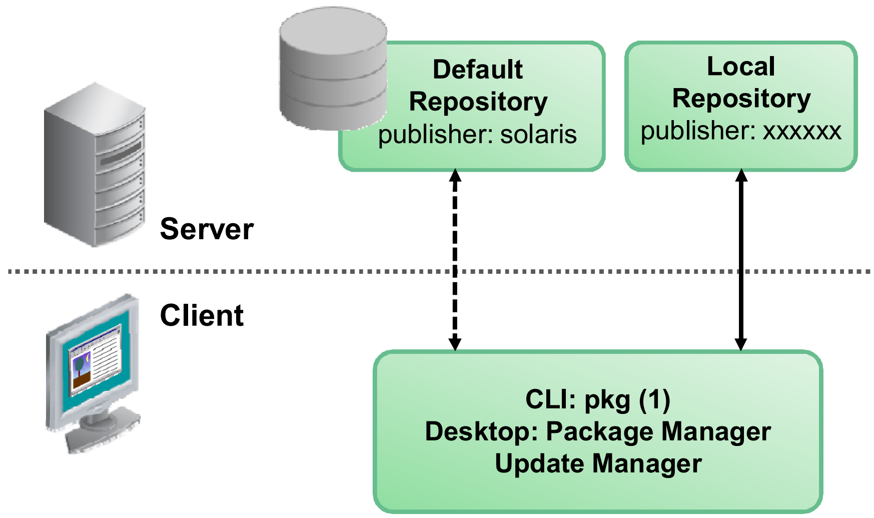
Other reasons your company might want to have a local copy of a package repository include:

- **Performance:** Having a local package repository provides client with access to packages at local network speeds.
- **Security:** You might not want your client systems to have access to the Internet.
- **Replication:** You want to make sure that an installation that you perform next year is exactly the same as the installation you perform today.

Oracle's default repository provides a complete archive of software packages to allow you to set up a local network IPS repository to which client systems can connect.

Note: Configuring a local IPS package repository is covered in the *Oracle Solaris 11 Advanced System Administration* course.

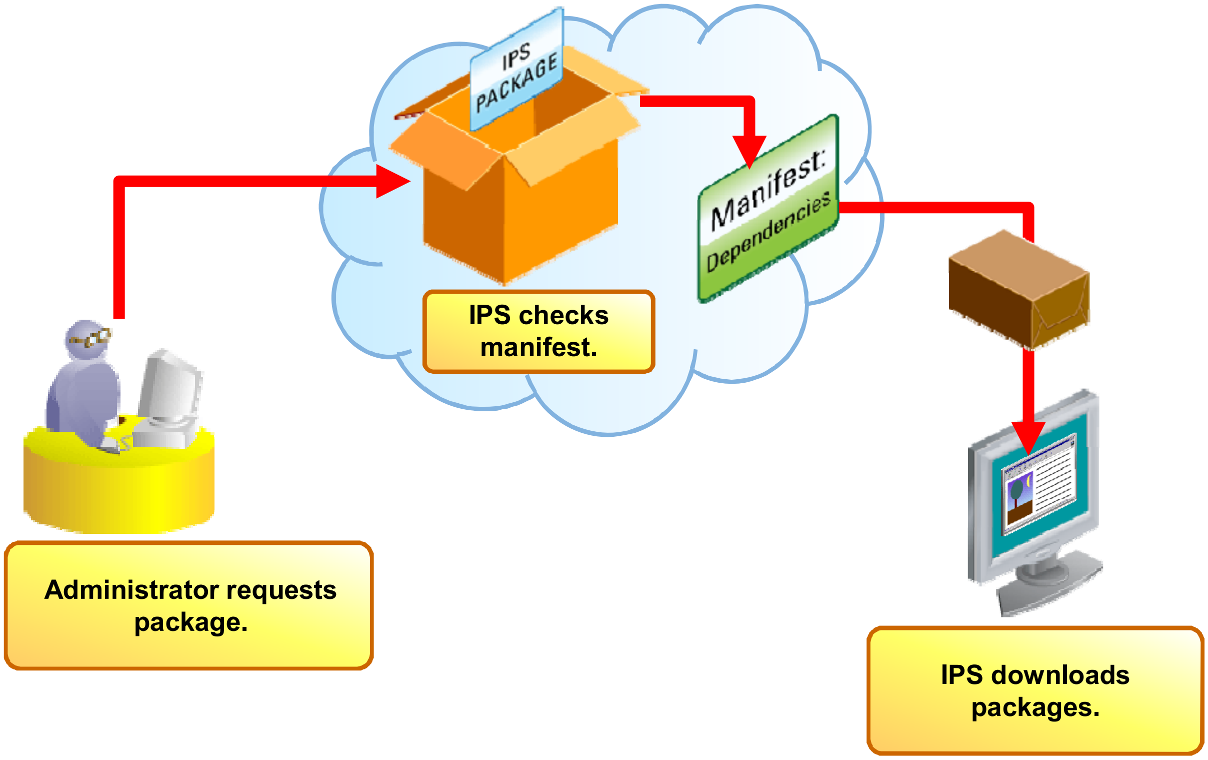
Software Update Process: Overview



An IPS repository contains collections of software packages. All packages within an IPS package repository reside in a catalog. The packages in a catalog are associated with a specific publisher. A publisher is a domain name that identifies a person, group of persons, or organization that develops and makes available one or more packages.

When you install Oracle Solaris 11, the system has only one publisher configured: the `solaris` publisher. You can, however, configure the system to support multiple publishers.

Installing and Managing Packages



When a system has been configured to interface with IPS, you can use IPS commands to install and manage the software packages on the operating system. For example, to install packages, you send a request from the client system to an IPS repository.

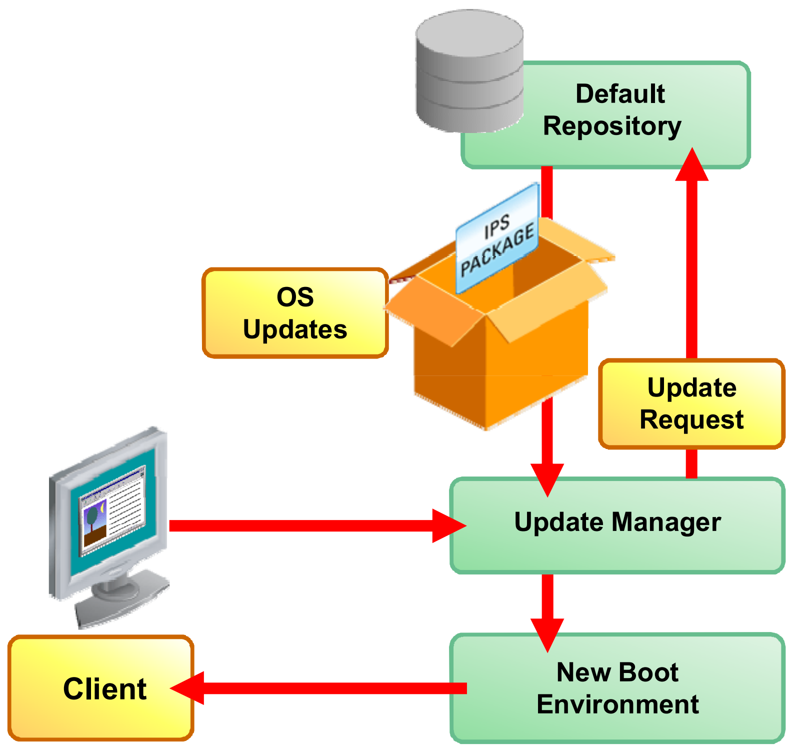
IPS looks for the software packages from the designated publisher. When IPS finds the software packages, it performs a dependency check on each package by checking the manifest that is included.

Note: The manifest describes the components and attributes that make up the package.

IPS finds all the packages in the list of dependencies, downloads these packages as well as the packages you requested to the client system, and installs them.

After IPS has installed the packages to the client system, you can manage them. You can list, search, and uninstall packages as well as perform a variety of other functions. You learn about the package management commands later in this lesson.

Updating the Operating System



In a console environment, package updates are performed using the package update command (`pkg update`), which you will look at shortly. If you are running in a graphical environment, you can also use an IPS feature called Update Manager to update all the installed packages on your system to the latest version. Update Manager automatically and continuously monitors for updates.

Note: This process is also referred to as *updating an image*. In IPS, an image is defined as a location where packages can be installed (for example, your Solaris instance).

During the update process, each package is updated by default from the publisher that provided the current installed version.

If particular packages are updated that affect the operating system's core programs, a new boot environment is created.

You now briefly examine boot environments.

Introducing Boot Environments

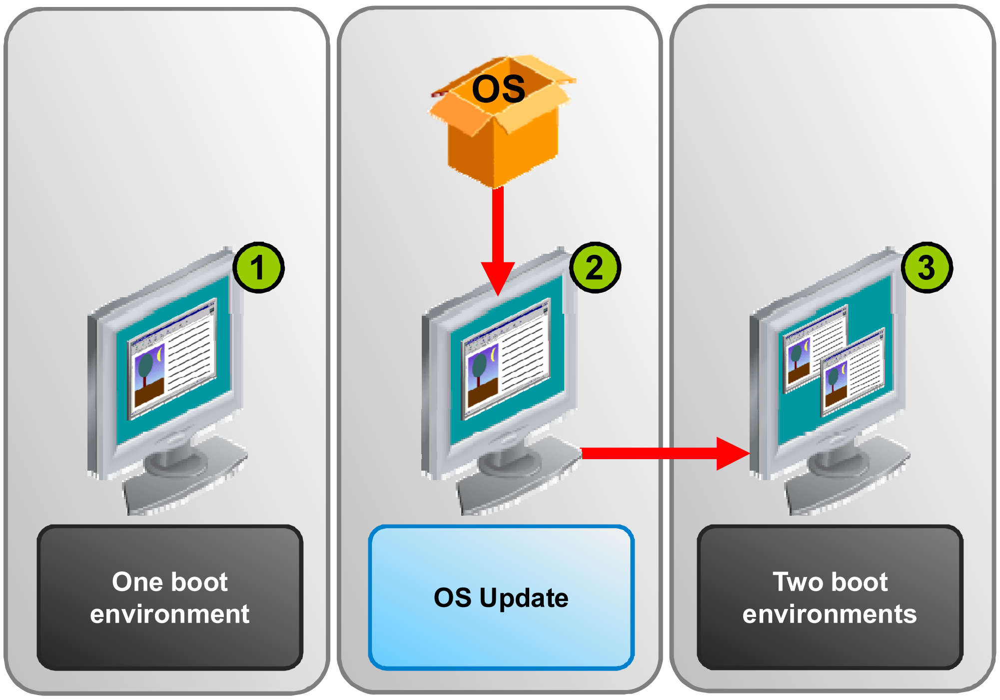
- A boot environment (BE) is a bootable instance of an Oracle Solaris 11 operating system image.
- Multiple boot environments can be maintained on a system.
- BEs make updating software a low-risk operation.
- BEs can have different software versions installed.

A boot environment (BE) is a bootable instance of an operating system image. That is, a BE contains a version of the operating system that can be started and is functional.

You can maintain multiple BEs on your system, and each BE can have different software versions installed. When you boot your system, you have the option to boot in to any of the BEs on the system.

With multiple boot environments, the process of updating software becomes a low-risk operation. System administrators can create backup boot environments before making software updates to their system. Administrators have the option of booting a backup boot environment if necessary.

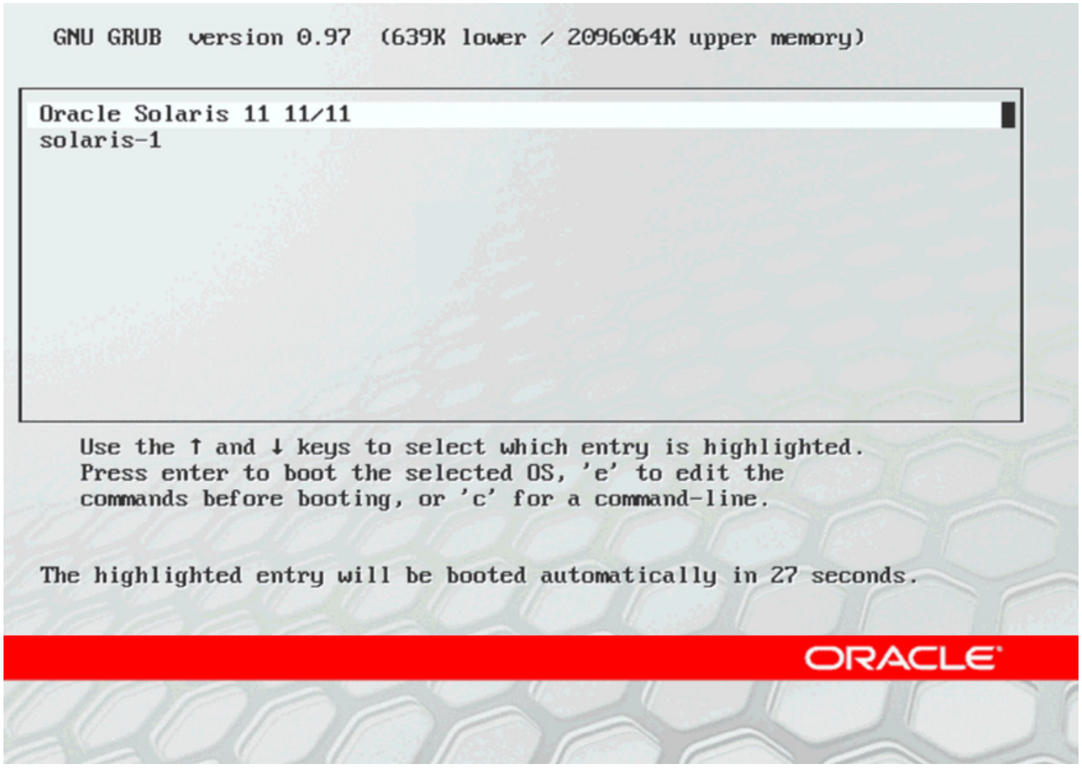
New Boot Environment Creation: Example



For example, when you first install the Oracle Solaris operating system, a new BE is automatically created. If you then run the update function on this image to update all the installed packages to the latest version, a new BE is created. The system sets this new BE as the default boot choice the next time the system is booted. The original or current BE remains as an alternative boot choice.

The result is that you now have two BEs: the new BE contains the most current version of the operating system and the alternative BE contains an older version of the operating system. Having the alternative boot environment enables you to return to that version of the operating system if you encounter issues with the new version.

Introducing Boot Environments



This screen capture of a GNU GRUB menu shows a system with two boot environments. The boot environment that is highlighted is the default or active boot environment. The other boot environment is an inactive, alternative boot environment. Later in this lesson, you learn how to manage systems that have multiple boot environments.

On the job, your company should tell you what the policy is for maintaining BEs and how to manage them.

Introducing the IPS Interfaces

IPS supports the following interfaces:

- Command-line
- GUI
  - Package Manager
  - Update Manager

With IPS, you have the option of using a command-line interface (CLI), graphical user interface (GUI), or web browser to perform package management tasks. IPS supports two GUIs: Package Manager and Update Manager. You will learn how to use the CLI commands and GUIs in the next section, and you will have the opportunity to work with them in the practices.

Implementing the Software Update Plan

Your assignment is to test:

- IPS update functionality by using:
  - Command-line interface
  - GUI interfaces
- Package management commands

As part of your company's preparations to implement Oracle Solaris 11 and as part of its software update test plan, you have been assigned to update the two test systems you installed in the previous lesson to the latest software release. Other members of your team have configured a local IPS repository. This repository contains the latest software packages from Oracle's default repository. Your task is to test the IPS update functionality by using both the command-line and GUI interfaces. After updating the OS successfully on both your test systems, you test the package management commands against the packages installed on the system.

In the next section, you look at the commands for performing these tasks.

Quiz

With IPS, publishers publish packages to:

a. Boot environments
b. Images
c. Repositories

Answer: c

Quiz

IPS supports only one publisher and one repository.

a. True
b. False

Answer: b

Quiz

A new boot environment is created automatically every time a new package is installed.

a. True

b. False

Answer: b

Lesson Agenda

- Planning for an Oracle Solaris 11 OS Software Update
- Updating the Oracle Solaris 11 OS
- Managing Software Packages
- Administering Boot Environments

Updating the Oracle Solaris 11 Operating System

Update Manager:

- Is used to update all installed packages to the latest version
- Can be invoked in several ways:
  - CLI command
  - Package Manager GUI

As discussed in the previous section, IPS has a feature called Update Manager that enables you to update all the installed packages on your system to the latest version.

You can invoke Update Manager through the command-line interface or Package Manager GUI. In this section, you learn how to use both options to update the operating system.

Performing the Update with the Command-Line Interface (CLI)

To update the operating system with the CLI:

1. List the current package publishers.
2. Run the `pkg update` command.

```
# pkg publisher
PUBLISHER              TYPE     STATUS      URI
solaris                origin   online      http://pkg.oracle.com/solaris/release/

# pkg update
```

ORACLE

Before you invoke Update Manager, you should make sure that all the publishers that provide you with software updates are available from your system. From the CLI, you can run the command `pkg publisher` to view the list of currently available publishers.

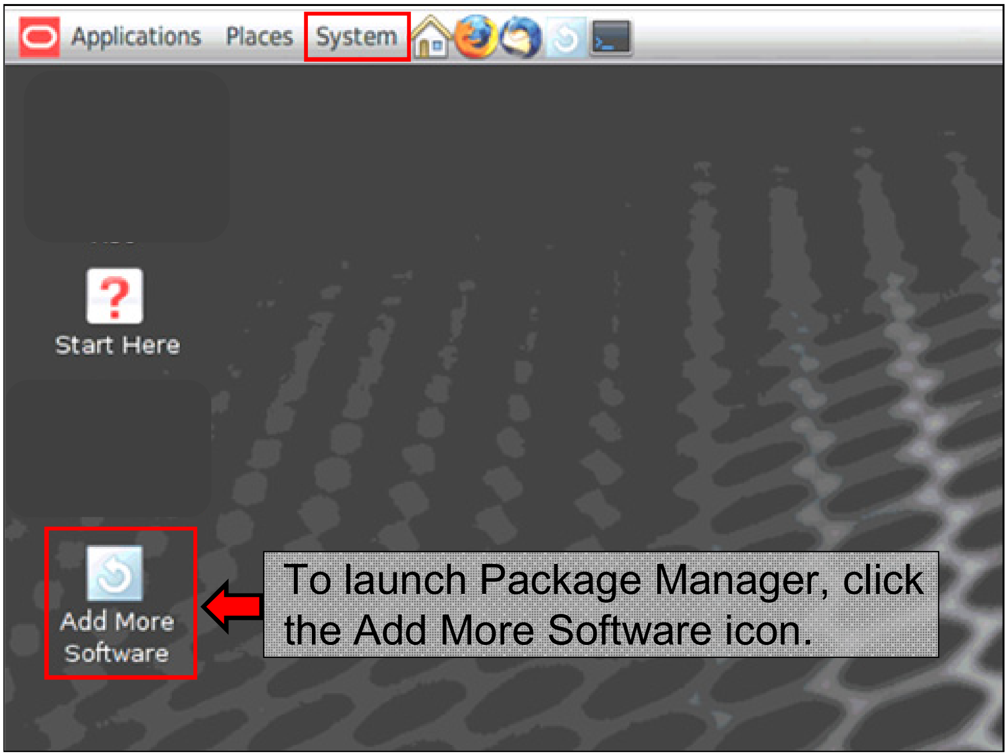
When you have verified which publishers' repositories IPS will search for software updates, you can execute the `pkg update` command to invoke Update Manager and start the update process.

Note: The functionality associated with the `pkg update` command might change with the final release.

If no updates are available, the following message is displayed: `No update available for this image`.

Now you look at how to run the update from Package Manager.

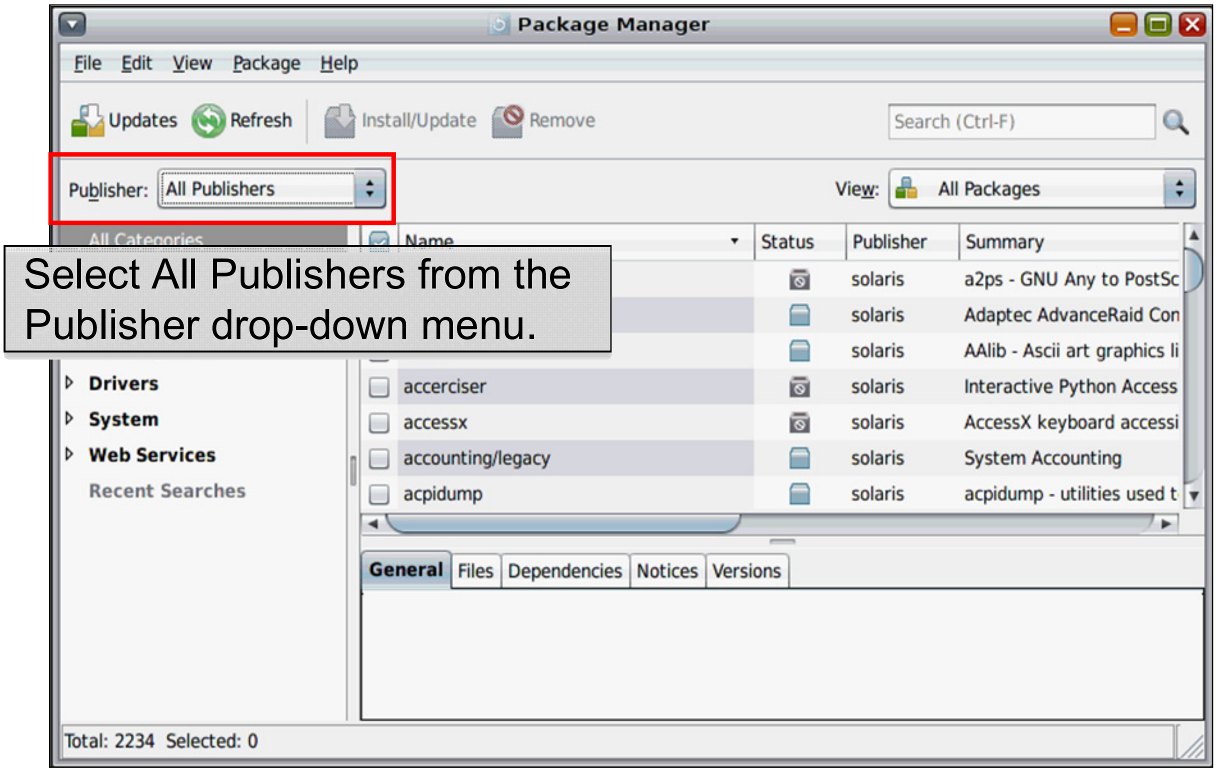
Performing the Update with Package Manager



From a desktop environment, you can invoke Update Manager either directly from the System menu item on the toolbar (System > Administration > Update Manager) or from the Package Manager GUI. To access Package Manager, click the Add More Software icon.

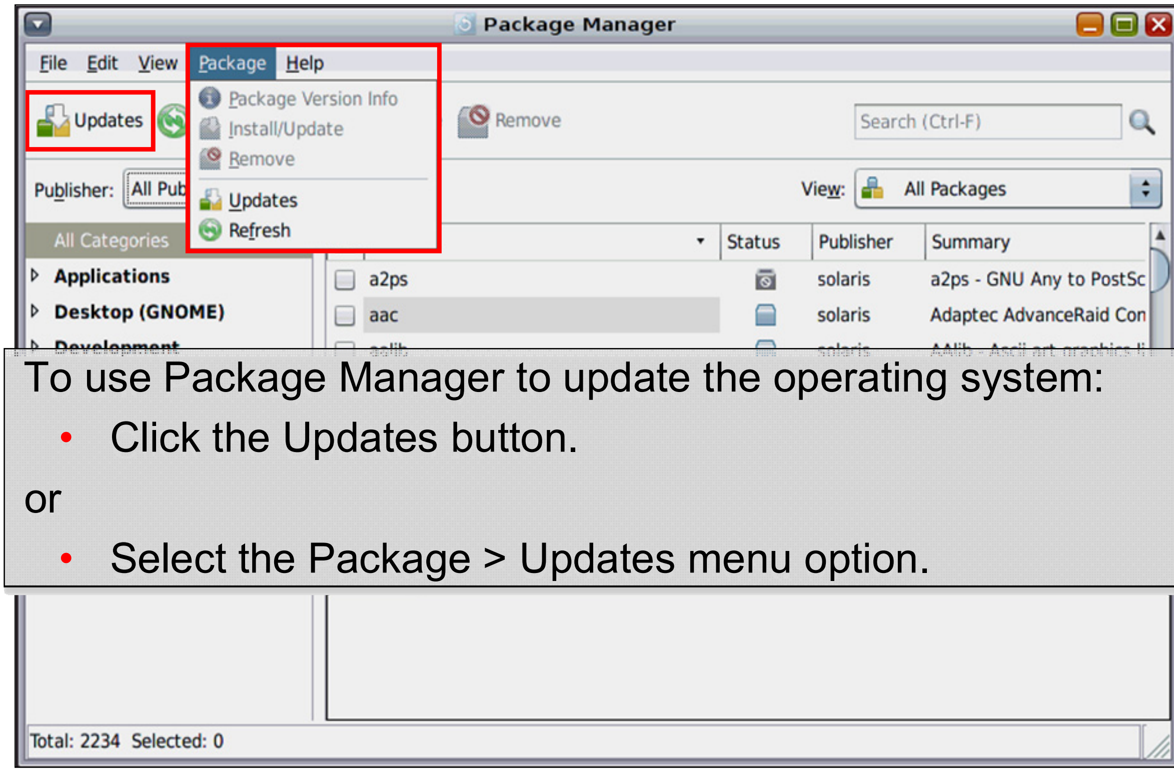
Note: You can also access Package Manager from the command line: `/usr/lib/pm-launch packagemanager`. For more information about the Package Manager command-line options, see the Oracle Solaris 11 IPS documentation.

Performing the Update with Package Manager



When Package Manager appears, you first select All Publishers from the Publisher drop-down menu located in the top-left portion of the GUI. This ensures that you are retrieving the latest software package updates from all the publishers that have provided your system with software.

Performing the Update with Package Manager



To use Package Manager to update the operating system:

- Click the Updates button.

or

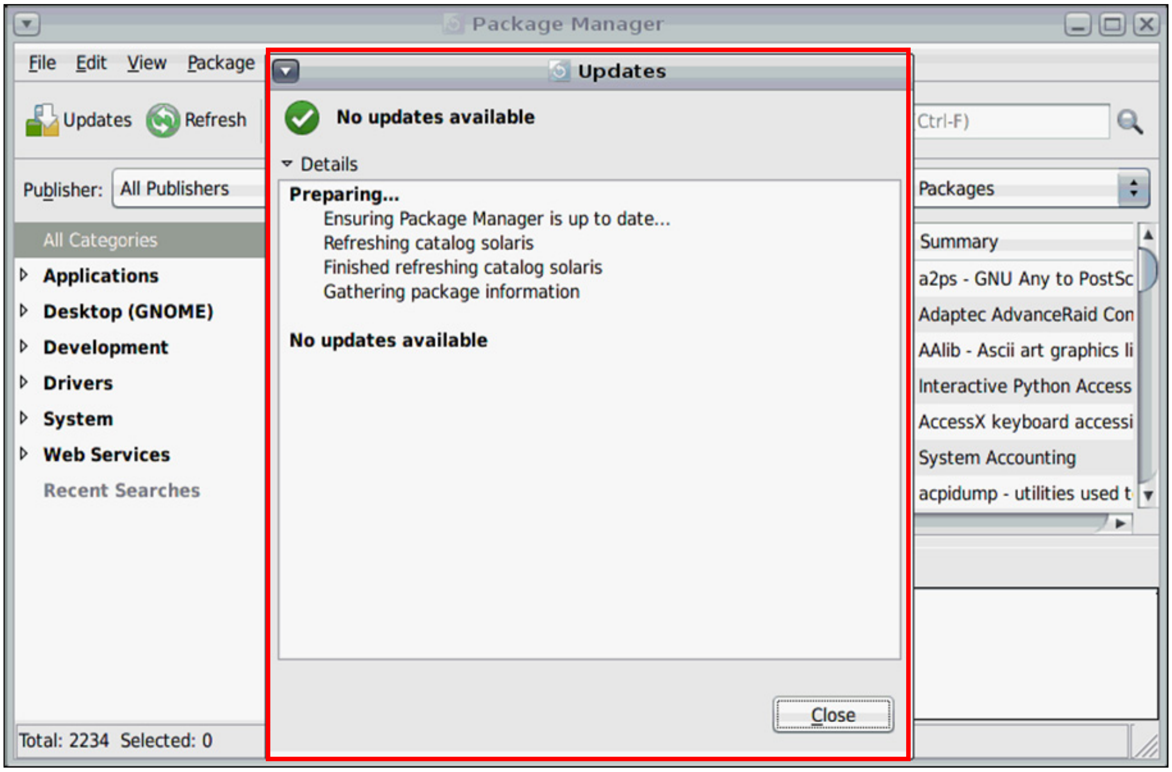
- Select the Package > Updates menu option.

When you have set the Publisher setting, you can access Update Manager and start the update process from Package Manager by doing either of the following:

- Click the Updates button on the toolbar of the GUI.
- Select Package > Updates from the menu bar.

The Updates window appears (see the next slide).

Monitoring the Update with Package Manager



IPS runs through a series of checks before it actually starts the update process. The first thing IPS does is make sure that Package Manager is up to date by accessing the catalogs for each of the publishers, which will identify any available updates. A catalog is all the packages in an IPS package repository. The packages in a catalog are associated with a specific publisher. After IPS refreshes the catalogs, it begins the update process by gathering package information.

If no updates are available, the following message is displayed: `No updates available`.

Rebooting the System

- If a new BE was created, do the following:
  - Edit the default BE name.
  - Click the Restart Now button.
- The new BE becomes the default boot environment.
- The current BE is available as an alternative boot choice.

If the system created a new boot environment (BE) for the update, you edit the default BE name and then click the Restart Now button to restart your system immediately. Click the Restart Later button to restart your system at a later time.

You must restart to boot in to the new BE. The new BE becomes your default boot environment. Your current BE is available as an alternative boot choice.

Practice 3-1 Overview: Verifying Access to the IPS Server

This practice covers verifying that the desktop client can access the local package repository.

In the practices for Lesson 3, you are presented with four tasks designed to reinforce the concepts presented in the lecture portion of this lesson. You will have the chance to perform the following tasks:

- **Practice 3-1:** Verifying access to the IPS server
- **Practice 3-2:** Managing software packages by using the command-line interface
- **Practice 3-3:** Managing software packages by using Package Manager
- **Practice 3-4:** Administering boot environments

You will find Practice 3-1 in your *Activity Guide*. It should take you about 15 minutes to complete.

Lesson Agenda

- Planning for an Oracle Solaris 11 OS Software Update
- Updating the Oracle Solaris 11 OS Using IPS
- Managing Software Packages
- Administering Boot Environments

Managing Software Packages

- Listing packages
- Displaying package information
- Installing and updating packages
- Viewing a package installation action without installing
- Verifying a package installation
- Searching for a package
- Uninstalling a package

This section focuses on managing the packages that have been installed on the system by using both command-line and GUI interfaces. You learn how you can:

- Find out what packages are currently on the system
- Search for packages
- Display information about packages, their contents, and publishers
- Install, update, and uninstall packages

It also covers how to view a package installation action without installing and how to verify a package installation.

You now run through each task first by using the command-line interface.

Listing Package State Information with the CLI

To list package state information, run `pkg list` *`pkg-fmri`*.

```
# pkg list entire
NAME (PUBLISHER)       VERSION            IFO
entire                 0.5.11-0.175       i--
```

ORACLE

To list the package state information, use the `pkg list` command with the package Fault Management Resource Identifier (FMRI), as shown in the example in the slide.

Note: The FMRI includes descriptive information about the package, such as the package name, version information, and date. For example, the FMRI pkg://solaris/developer/apptrace@0.5.11,5.11-0.175.0.1:20101104T230706Z consists of the following information:

- **Scheme:** pkg
- **Publisher:** solaris
- **Category:** developer
- **Package Name:** apptrace
- **Component Version:** 0.5.11
- **Build Version:** 5.11
- **Branch Version:** 0.175.0.1
- **Timestamp** (when the package was published): 20101104T230706Z The `pkg list` command shows whether an update exists for a package, whether an update can be installed in this image, and whether a package is obsolete or renamed.

The output of the `pkg list` command displays the following information about the package:

- **NAME (PUBLISHER):** Name of the package. If the publisher listed is not the preferred publisher, the publisher name is listed in parentheses.
- **VERSION:** Release and branch versions of the package
- **IFO:** Flags that indicate whether a package is installed, frozen, or obsolete/renamed

Displaying Package Information with the CLI

To display package information, run `pkg info pkg-fmri`.

```
# pkg info -r apptrace
          Name: developer/apptrace
       Summary: Apptrace Utility
   Description: Apptrace utility for application tracing,
    including shared objects
      Category: Development/System

         State: Installed
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
        Branch: 0.175.0.0.0.2.1
Packaging Date: November 19, 2011 05:30:54 AM
          Size: 159.64 kB
          FMRI: pkg://solaris/developer/apptrace@0.5.11,5.11-
    0.175.0.0.0.2.1:20111019T053054Z
```


To display information about a package, use the `pkg info` command with the package FMRI. In the example, information about the `apptrace` package is displayed. This example uses the `-r` option to display information even if the package is not installed.

The output of the `pkg info` command displays the following information about the package:

- Name
- Summary, to include the name and version
- State (for example, installed or not installed)
- Publisher
- Version
- Build release
- Branch
- Packaging date
- Size
- FMRI

Note: To view all the packages installed on the system, use `pkg info` without a package FMRI.

Displaying the Contents of a Package with the CLI

To display package contents information, run `pkg contents` *`pkg-fmri`*.

```
# pkg contents compress/zip
PATH
usr
usr/bin
usr/bin/zip
Usr/bin/zipcloak
Usr/bin/zipsplit
Usr/share
Usr/share/man
<output omitted>
```


To display the contents of a package, use the `pkg contents` command with the package FMRI. In the example, the contents for the `compress/zip` package are shown.

Note: By default, only the path attribute is shown. For information about the options that you can use with the `pkg contents` command, see the `pkg(1)` man page.

Updating an Installed Package with the CLI

To update an installed package, run `pkg update` *`pkg-fmri`*.

```
# pkg update compress/zip
No updates available for this image.
```

ORACLE

If the package is already installed, the package is updated by installing the newest version of the package by using the `pkg update command` with the package FMRI, as shown in this example.

Note: The functionality of this command might change with the final release of the software.

The system will try to update to the newest version possible within the current constraints of the system. The results of the update function might not always provide the expected results (for example, because of dependencies that require versions to be kept within a certain range).

The output of the `pkg update` command displays the following:

- Status of the download
- Number of packages that were updated
- Number of files that were updated
- Size of the download (in megabytes)

Installing a Package with the CLI

To install a package, run `pkg install` *`pkg-fmri`*.

```
# pkg install apptrace
               Packages to install:     1
           Create boot environment:    No
    Create backup boot environment: No

DOWNLOAD                         PKGS         FILES    XFER (MB)
Completed                        1/1          10/10    0.1/0.1

PHASE                                        ACTIONS
Install Phase                                  29/29

PHASE                                          ITEMS
Package State Update Phase                        1/1
Image State Update Phase                          2/2
```


To install a software package from an IPS package repository, use the `pkg install` command with the package FMRI (as shown in the example).

With a package update, by default, the newest version of a package is installed from the preferred publisher. The `install` subcommand installs the package if the package is not already installed. If the package is already installed, the package is updated by installing the newest version of the package.

Note: You can also control which publisher provides a package, as well as which version of the package you want to install. For more information about these options, see the Oracle Solaris 11 IPS documentation.

The output of the `pkg install` command displays the following:

- Status of the download
- Number of packages that were installed
- Number of files that were installed
- Size of the download (in megabytes)

Viewing an Installation Action Without Installing with the CLI

To view an installation action without installing, run `pkg install -n` *`pkg-fmri`*.

```
# pkg install -nv apptrace
           Packages to install:         1
     Estimated space available:  25.45 GB
Estimated space to be consumed:  15.76 MB
       Create boot environment:        No
Create backup boot environment:        No
          Rebuild boot archive:        No
…
…
…
```


If you want to check exactly what is going to be installed before you actually install a software package, you can use the `-nv` option with the `pkg install` command, as shown in the example.

Note: You can also use the `-n` option with the `update` subcommand.

Verifying a Package Installation with the CLI

To verify a package installation, run `pkg verify pkg-fmri`.

```
# pkg verify -v apptrace
Verifying: PACKAGE                                      STATUS
pkg://solaris/developer/apptrace                        OK
```


To verify or validate that the package installed correctly, use the `pkg verify -v` command followed by the package FMRI.

Note: If you want to run a `verify` against more than one package, you can specify more than one *`pkg-fmri`* pattern. If you want to display more verbose information messages, you can use the `-v` option, and to display only error messages, you can use the `-q` option.

Searching for a Package with the CLI

To search for a package, run `pkg search` *`pattern`*.

```
# pkg search -l bash
INDEX       ACTION VALUE                    PACKAGE
basename    dir    etc/bash                 pkg:/shell/bash@4.1.9-0.175.0.0.0.2.537
basename    dir    usr/share/bash           pkg:/shell/bash@4.1.9-0.175.0.0.0.2.537
basename    file   usr/bin/bash             pkg:/shell/bash@4.1.9-0.175.0.0.0.2.537
pkg.fmri    set    solaris/shell/bash       pkg:/shell/bash@4.1.9-0.175.0.0.0.2.537
```


To search for a package, use the `pkg search` command with the package FMRI. In this example, you are searching for the bash package in the installed image.

Note: By default, matches are displayed only for currently installed or newer package versions. If you want to display all matched versions, you can do so by using the `-f` option.

The output of the `pkg search` command displays the following:

- **INDEX:** Where in the data the match was found
- **ACTION:** What kind of action matched the query. In the example, a set action, a file action, and two directory actions matched the query.
- **VALUE:** Directory in which the match was found
- **PACKAGE:** Package FMRI for the package that matched the query

Note: There are multiple ways you can use the `pkg search` command. For example, you can restrict your search to specific repositories or files, or you can use wildcards and Boolean directives. For more information about these options, see the Oracle Solaris 11 IPS documentation.

Uninstalling a Package with the CLI

To uninstall a package, run `pkg uninstall` *`pkg-fmri`*.

```
# pkg uninstall apptrace
                    Packages to remove:     1
               Create boot environment:    No
        Create backup boot environment:    No

PHASE                                            ACTIONS
Removal Phase                                      26/26

PHASE                                              ITEMS
Package State Update Phase                           1/1
Package Cache Update Phase                           1/1
Image State Update Phase                             2/2
```


To uninstall a package, use the `pkg uninstall` command with the package FMRI, as shown in this example.

Note: You can also use the `-n` option with the `uninstall` subcommand if you want to check what changes would occur without actually performing the uninstall.

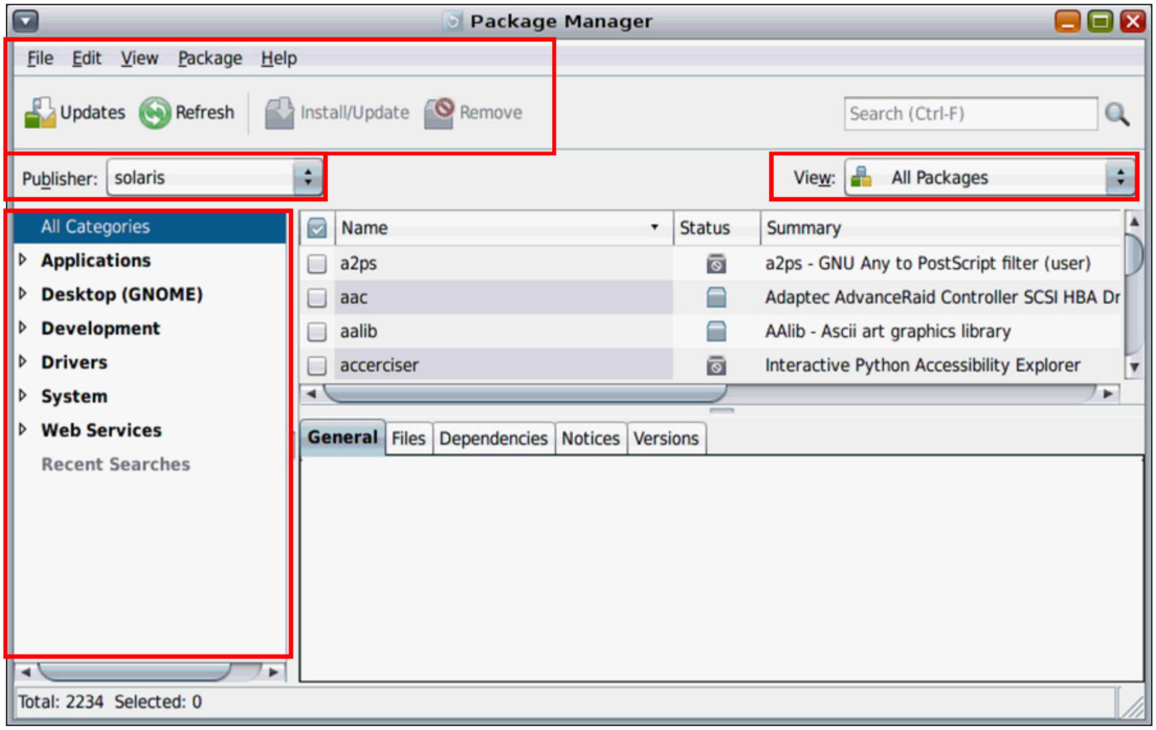
Package Management Commands: Summary

Package Management Task	IPS Command
Display package state and version information	`pkg list`
Display package information	`pkg info`
Display contents of a package	`pkg contents`
Install package updates	`pkg update`
Install package	`pkg install`
Verify package installation	`pkg verify`
Search for a package	`pkg search`
Uninstall a package	`pkg uninstall`

The table in the slide lists, by task, the IPS package management commands presented in this lesson. You now learn how to perform these tasks by using Package Manager.

Note: As you look at Package Manager, keep in mind that although Package Manager supports many of these same package management tasks, it organizes and displays the package information in slightly different ways.

Managing Packages with Package Manager



The Package Manager GUI has a robust interface that allows you to list, search, install, update, and remove packages. You can perform these actions in a number of ways. In the top-left corner of the interface is a menu bar. You can use the options presented on this menu bar as follows:

- **File:** Add and manage publishers and boot environments
- **Edit:** Perform various edit functions and set preferences
- **View:** View Package Manager logs
- **Package:** View package information, and perform the same package management tasks that are offered by the Updates, Refresh, Install/Update, and Remove buttons that are located just below the menu
- **Help:** Access information about how to use Package Manager

Just below the menu bar is a row of the following buttons:

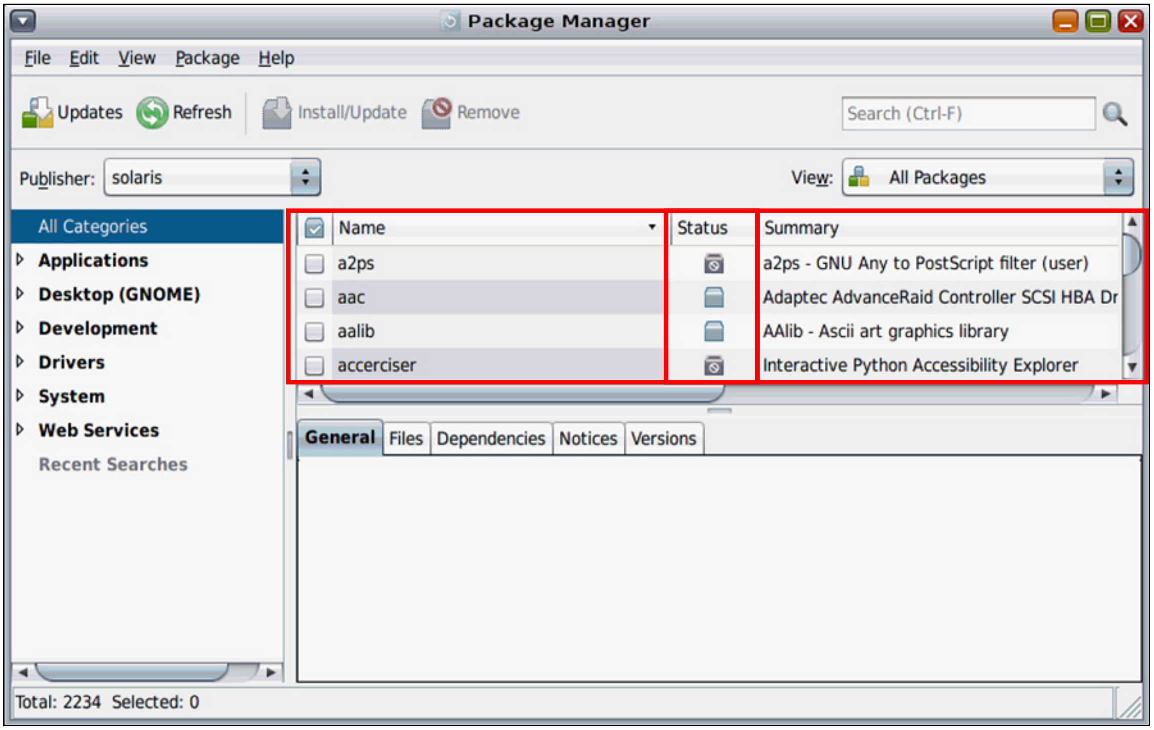
- **Updates:** Enables you to retrieve updates (as we saw in the section on updating all the packages installed on the system)
- **Refresh:** Enables you to refresh the GUI
- **Install/Update:** Enables you to perform the install/update function on a selected package or packages
- **Remove:** Uninstalls a selected package or packages

You have already looked at the Publisher drop-down menu that is located just below the row of buttons.

The View drop-down menu to the far right of the Publisher button enables you to view all packages, installed packages only, updates only, uninstalled packages, and packages you have selected.

Located below the Publisher menu is a list that enables you to decide, by category, how you want to view packages. For example, you can view all categories of packages or just application packages, desktop packages only, development packages only, and so on. Within each category, there are subcategories that enable you to restrict your list of packages further.

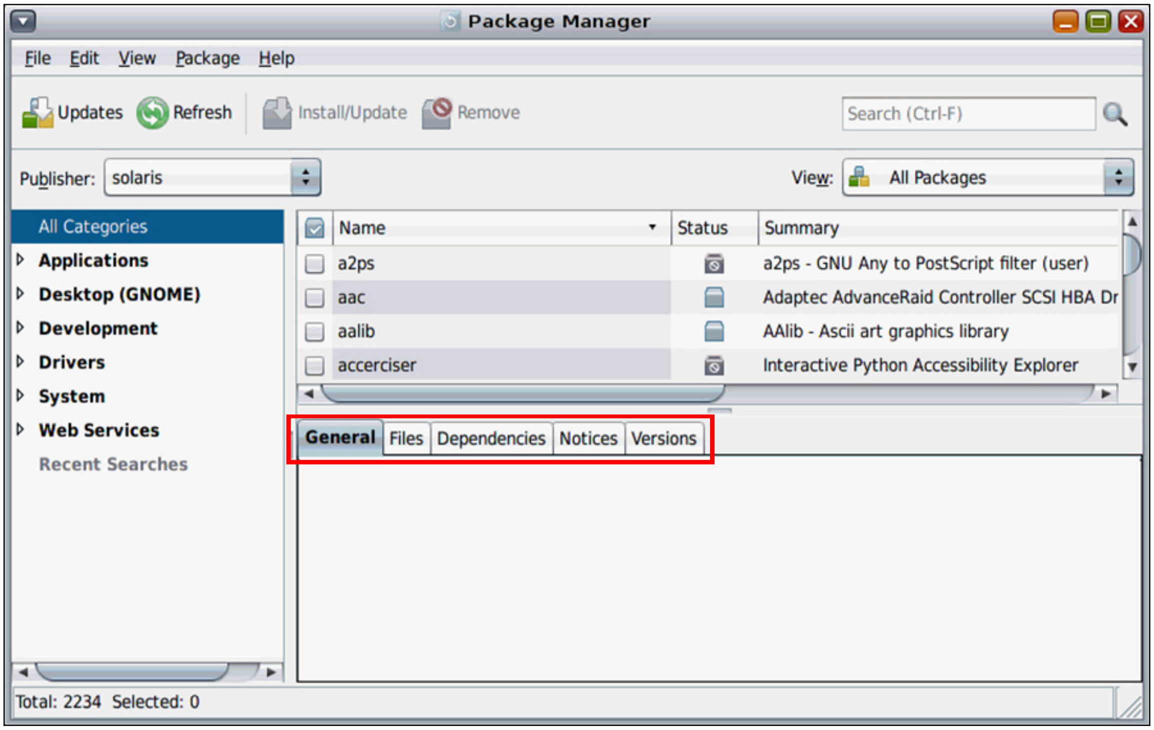
Managing Packages with Package Manager



After you have selected the category, the associated list of packages is displayed in the center of the interface, as shown in the highlighted area in the example.

The name, status, and summary are displayed for each package. If a package is installed, an “active” folder icon appears in the Status column; the folder is a solid light blue. If the package has not been installed, an “inactive” folder icon is displayed; the folder is disabled and has a “no” symbol displayed on it.

Managing Packages with Package Manager

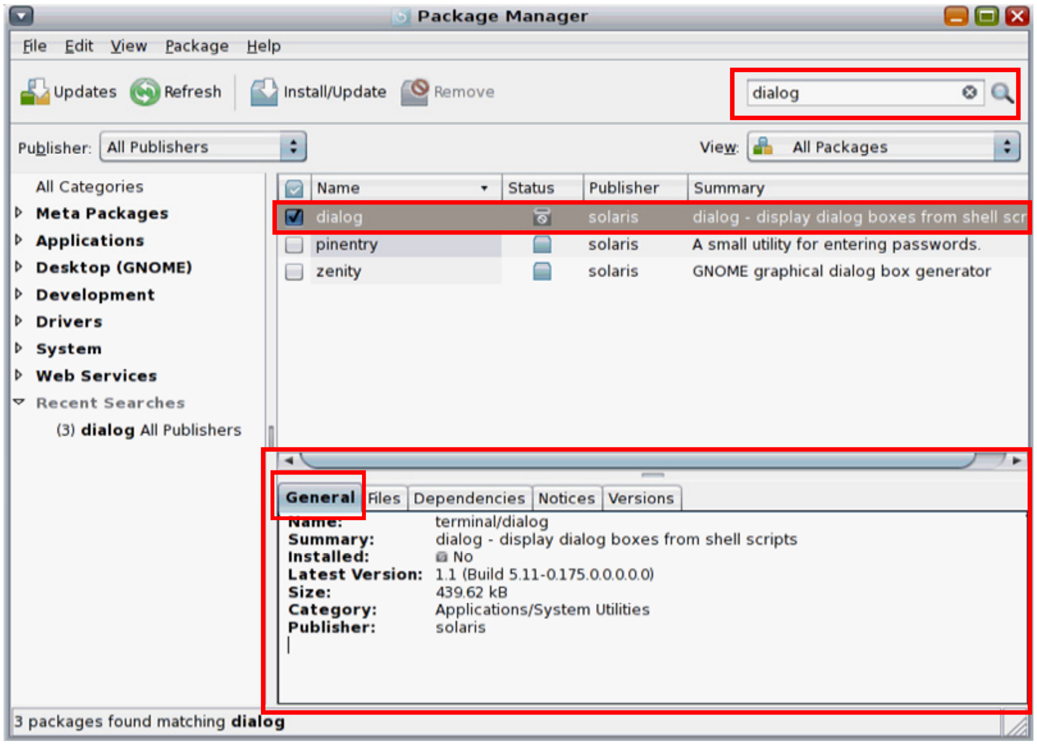


To see information that is specific to a file, you can use the tabs that appear below the list of packages. The tabs from left to right are:

- **General:** Displays information that is similar to the `pkg info` command output
- **Files:** Lists the files associated with the selected package
- **Dependencies:** Lists the other software packages by build and the installed version that the selected package requires in order to run
- **Notices:** Displays any messages pertaining to the selected package
- **Versions:** Displays the name of the package, the installed version, and any version to install, and an active Install/Update button if an updated version is available for download

Now that you are more familiar with the layout of the interface, you can look at how to perform specific package management tasks by using the dialog-box display package called `dialog`. Your first step is to search for the package.

Displaying Package Information with Package Manager



To search for the `dialog` package, enter a keyword in the search field and click the magnifying glass. If found, the package is displayed in the main window.

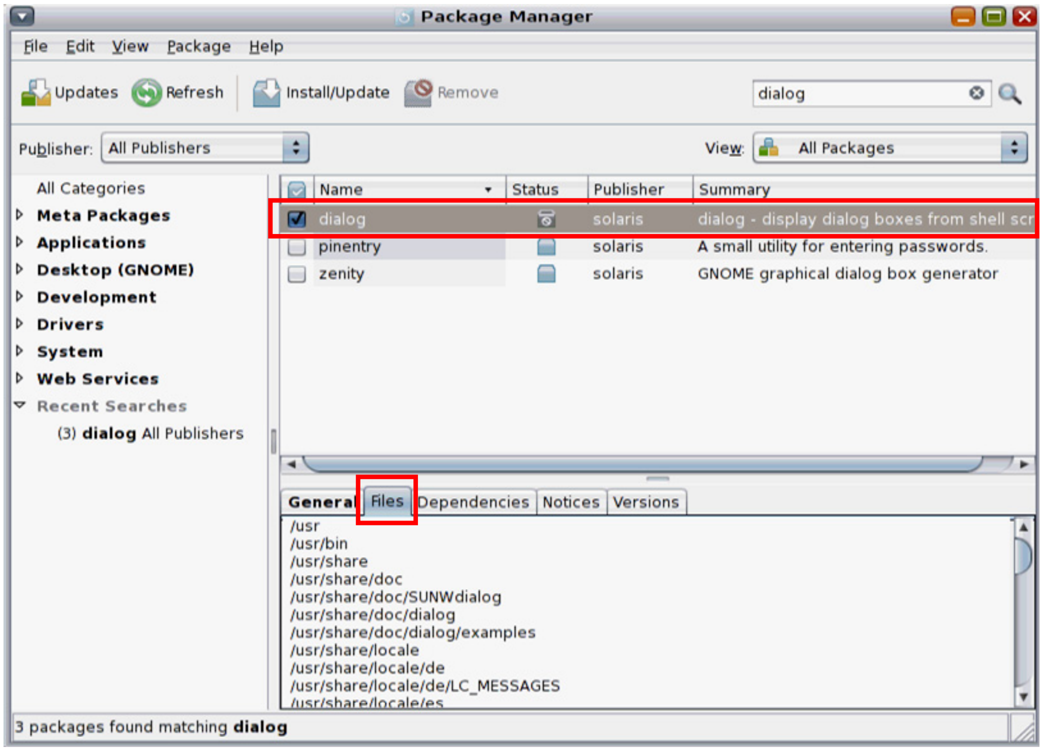
To display information about the package, select the package and click the General tab, as shown in the example.

The following information is displayed on the General tab:

- Name
- Summary (name)
- Installed (yes or no)
- Latest version
- Size
- Category
- Publisher

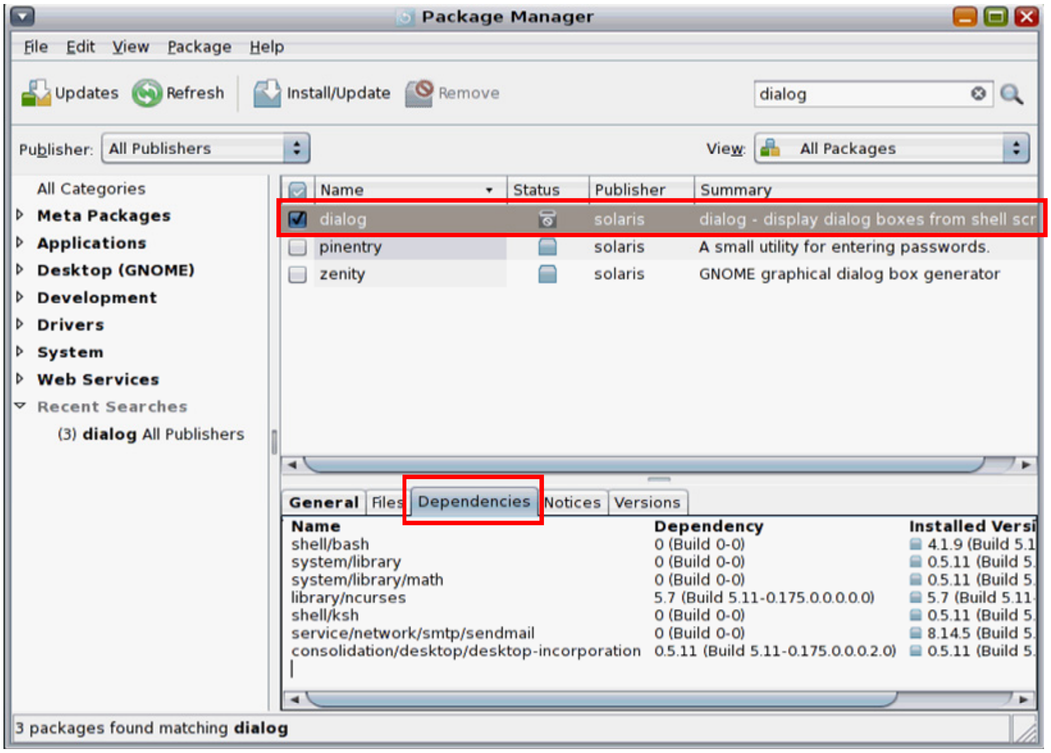
As you can see, the information differs slightly from the information displayed with the `pkg info` command.

Displaying the Files of a Package with Package Manager



With Package Manager, you can display the files of a package as well as its location by selecting the package and then clicking the Files tab (as shown in the example). Here, you can see the files called out in the `dialog` manifest.

Displaying Package Dependency Information with Package Manager

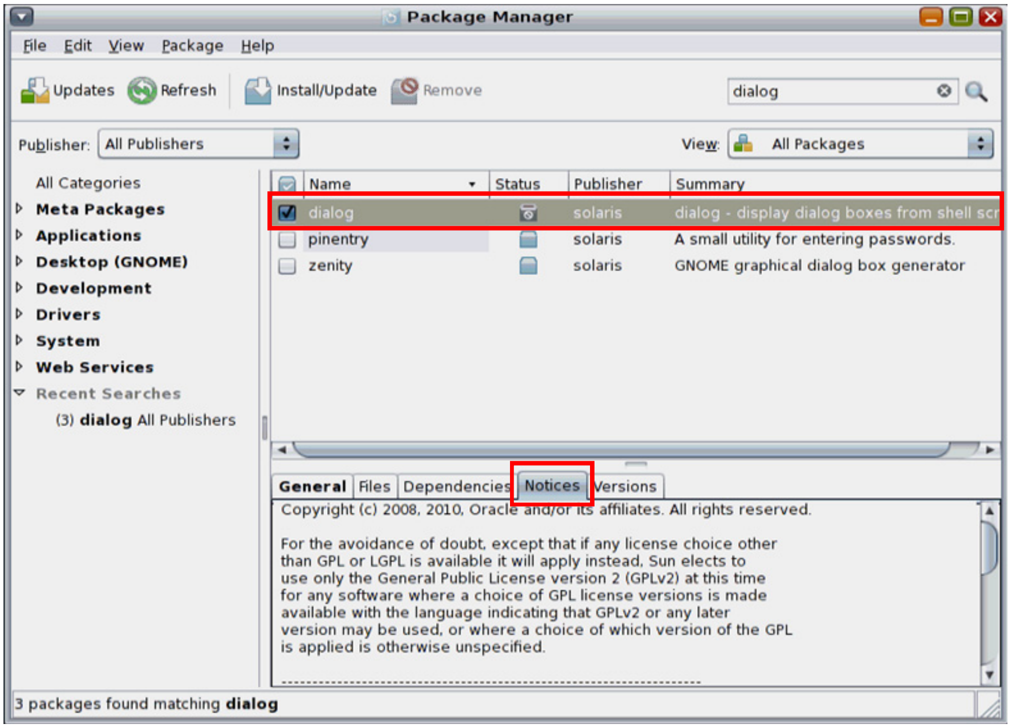


As you know from the discussion on updating the operating system, software packages are often dependent on having other packages installed in order to run properly. By using Package Manager, you can display all the package dependencies for a package by selecting a package and clicking the Dependencies tab, as shown in the example.

The following information is displayed on the Dependency tab:

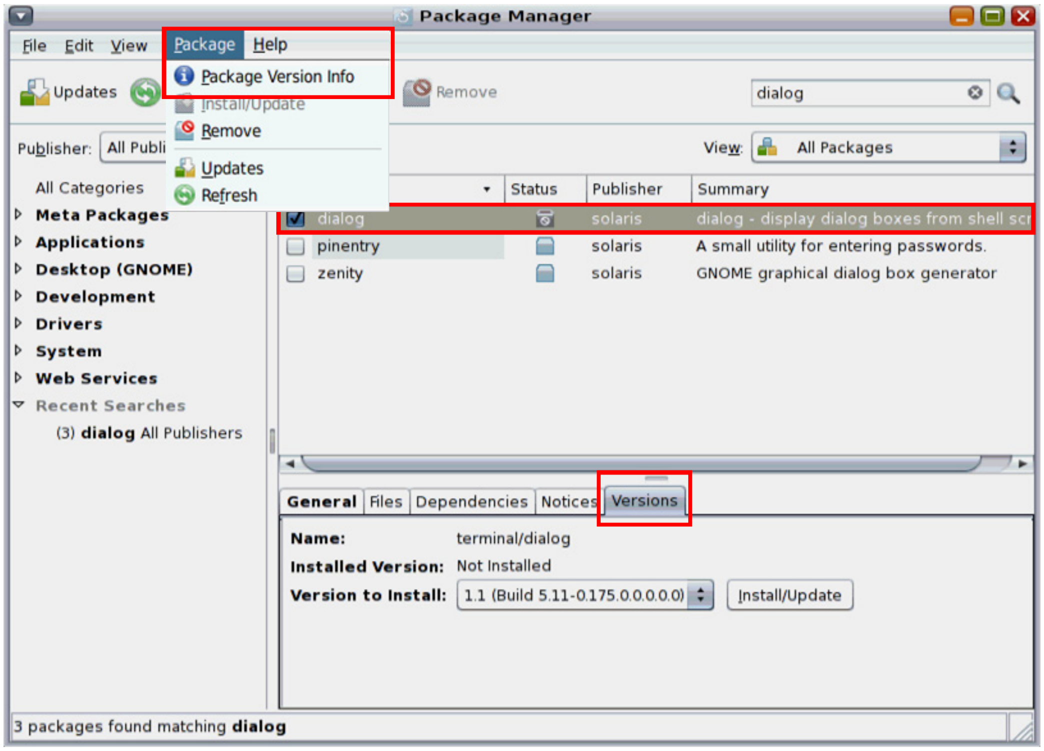
- **Name:** Name of the package
- **Dependency:** Build information
- **Installed Version:** The version of the package that is currently installed on the system

Displaying Package Notices with Package Manager



If any messages or notices have been generated in association with a package, Package Manager enables you to display these notices by selecting a package and clicking the Notices tab, as shown in the example.

Displaying Package Versions with Package Manager



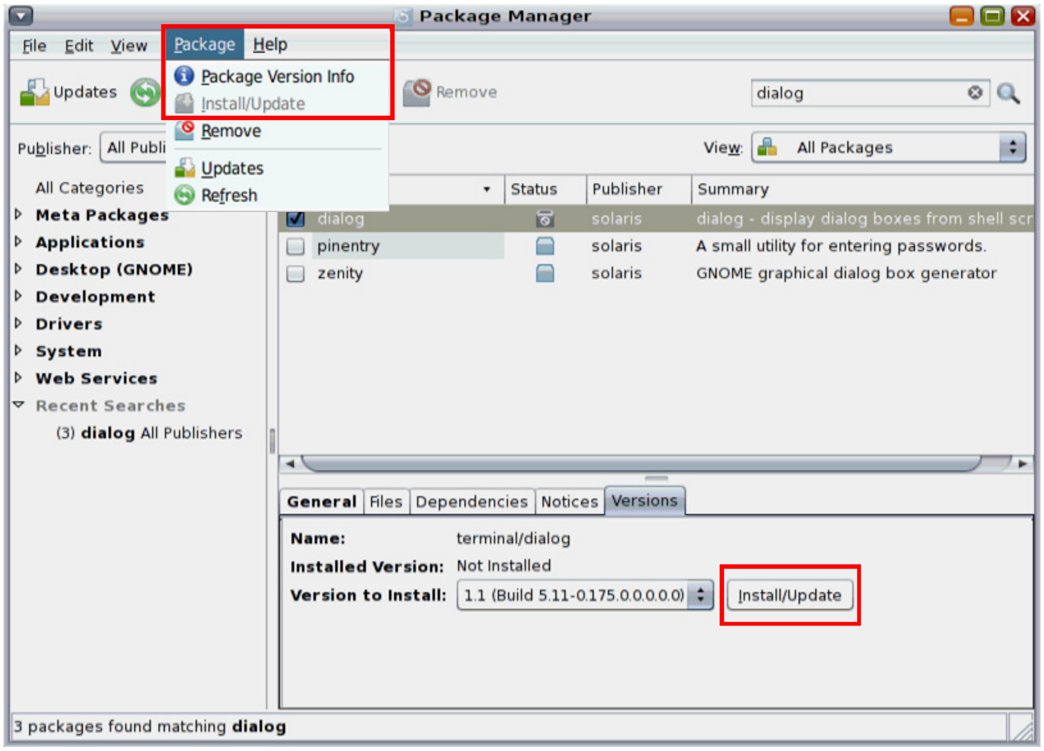
There are several ways you can display version information for a package with Package Manager. You can select Package > Package Version Info from the menu, as shown in the first highlighted section of the screenshot in the slide. Or you can select the package and click the Versions tab, as shown in the second highlighted example.

The following information is displayed on the Versions tab:

- **Name:** Name of the package
- **Installed Versions:** Version or versions currently installed on the system
- **Version to Install:** Indicates if an updated version of the package is available for installation. The Install/Update button is active if an updated version is available for download.

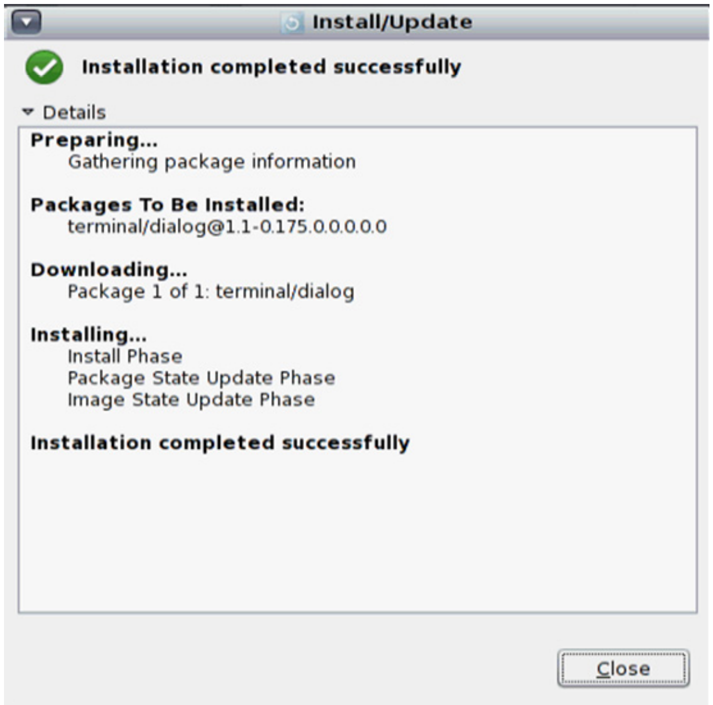
Note: The Package > Package Version Info option does not provide the Install/Update button.

Installing and Updating a Package with Package Manager



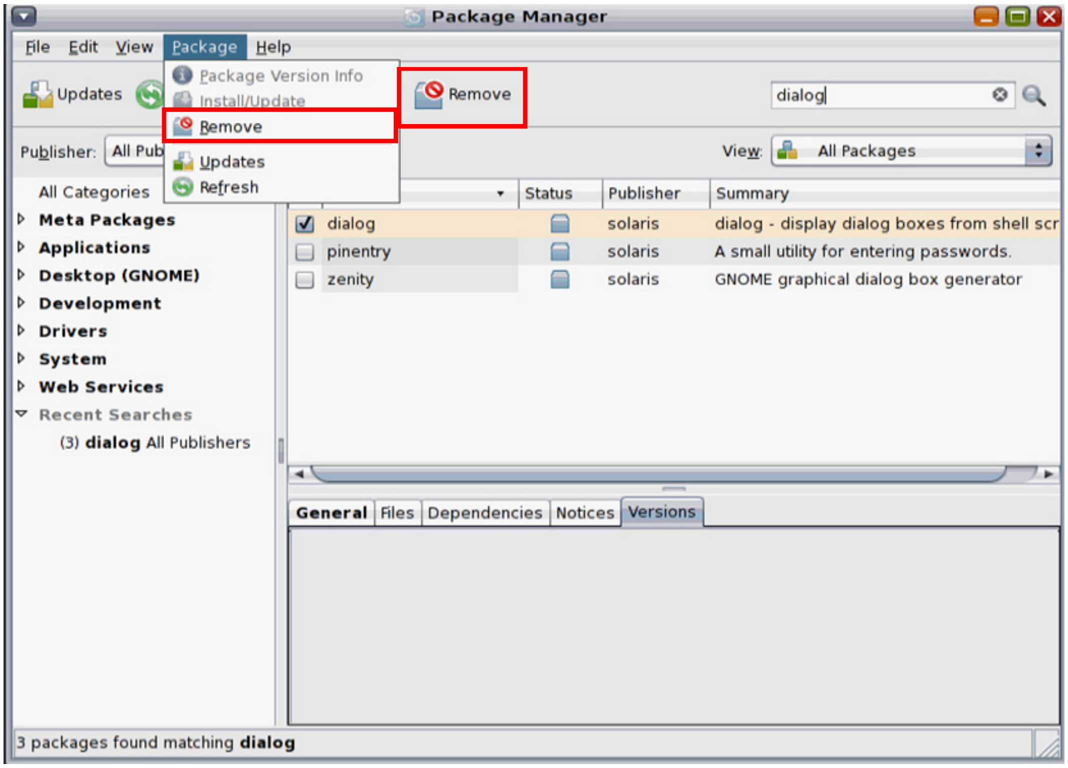
To install or update a package with Package Manager, you can either select Package > Install/Update from the menu or click the Install/Update button. As you can see in the example, there is a version of the package available for installation.

Verifying a Package Installation with Package Manager



When the installation completes, you are notified, as shown in the dialog box presented here. To close the dialog box and return to Package Manager, click the Close button.

Uninstalling a Package with Package Manager



To uninstall a package with Package Manager, you highlight the package and then either select Package > Remove from the menu, or select the package and click the Remove button.

Practice 3-2 and Practice 3-3 Overview: Managing Software Packages by Using the Command-Line Interface and Package Manager

These practices cover the following topics:

- Searching for a package
- Performing a test run on the package installation
- Installing a package
- Verifying the package installation
- Displaying information about the package and its contents
- Uninstalling a package

It should take you a total of about one hour to complete both these practices.

Lesson Agenda

- Planning for an Oracle Solaris 11 OS Software Update
- Updating the Oracle Solaris 11 OS Using IPS
- Managing Software Packages
- Administering Boot Environments

Administering Boot Environments

- Listing existing boot environments
- Creating a new boot environment
- Renaming an existing, inactive boot environment
- Activating an existing, inactive boot environment
- Destroying an existing, inactive boot environment

BE management utilities:
- `beadm` command
- Package Manager

Administering boot environments is part of a system administrator's responsibility. As you saw when updating the operating system, a new boot environment was automatically created, providing you with a new current boot environment and an alternative boot environment based on the older version of the operating system.

In this section, you look at the different ways you can administer boot environments and when and why you would do so. You learn about how to list the boot environments that exist on your system; create a new boot environment; and rename, activate, and destroy an existing, inactive boot environment.

IPS has two BE management utilities that you can use to perform these tasks:

- **`beadm` command:** Provides a full range of options for managing boot environments. Requires root privileges.
- **Package Manager:** Provides a subset of boot environment management options

You learn about how to use both utilities as each task is covered. You begin by examining how a new boot environment is created.

Listing the Boot Environments on the System

To list the boot environments on a system, run `beadm list`.

```
# beadm list
BE          Active  Mountpoint Space   Policy  Created
--          ------  ---------- -----   ------  -------
solaris      NR      /          4.87G  static  2011-11-19 20:38
solaris-1   -        -        121.39M  static  2011-11-23 06:59
solaris-2   -        -         146.0K  static  2011-11-23 07:31
```


Before you start managing the boot environments on a system, you should determine what boot environments exist on the system. To determine this, use the `beadm list` command, as shown in the example.

The `beadm list` command displays the following information:

- **BE:** Name of the boot environment
- **Active:** Boot status of the boot environment. In the example for the `solaris` boot environment, `N` means the boot environment is currently active and `R` means it will be the boot environment that will be active on reboot as well. A dash (`-`) indicates that the BE is currently inactive.
- **Mountpoint:** Where the boot environment is mounted. In the example, the `solaris` boot environment is mounted in the root (`/`) directory.
- **Space:** Size of the boot environment
- **Policy:** Either static or volatile
- **Created:** Date the boot environment was created

Note: There are a number of options that can be used with the `beadm list` command but that are outside the scope of this course. For more information about these options, see the Oracle Solaris 11 documentation for managing boot environments.

Creating a New Boot Environment with `beadm`

To create a new boot environment, run `beadm create` *`beName`*.

```
# beadm create test1
# beadm list
BE          Active  Mountpoint Space   Policy  Created
--          ------  ---------- -----   ------  -------
solaris      NR      /          4.87G  static  2011-11-19 22:14
solaris-1   -        -        121.39M  static  2011-11-23 06:59
solaris-2   -        -         146.0K  static  2011-11-23 07:31
test1       -        -         146.0K  static  2011-11-23 07:34
#
```

ORACLE

As you have seen, the system automatically creates a new boot environment as part of an update operation if the software packages that are being installed affect the core operating system files.

However, there may be times when you need to manually create a new boot environment. For example, you might want to create a backup of an existing boot environment prior to modifying the original boot environment for the purposes of testing a new application.

To create a new boot environment from the active boot environment, use the `beadm create` command with the name of the new boot environment, as shown in the example. You determine the name of the boot environment (for example, `test1`). The `beadm` command creates a new boot environment that is a clone of your active boot environment. This clone is inactive. If you run `beadm list` again, you should see your new boot environment listed.

Note: You can also create a boot environment from an inactive boot environment. For more information about how to perform this task, see the Oracle Solaris 11 documentation for managing boot environments.

Renaming an Existing, Inactive Boot Environment with `beadm`

To rename a boot environment, run `beadm rename` *`beName newBeName`*.

```
# beadm rename test1 apptest1
# beadm list
BE          Active  Mountpoint Space   Policy  Created
--          ------  ---------- -----   ------  -------
apptest1     -        -        146.0K  static  2011-11-23 07:34
solaris      NR       /          4.87G static  2011-11-19 22:14
solaris-1   -         -       121.39M  static  2011-11-23 06:59
solaris-2   -         -        146.0K  static  2011-11-23 07:31
#
```


You can rename an existing, inactive boot environment with `beadm` by using the `beadm rename` command with the current boot environment name, followed by the new boot environment name. In this example, the boot environment name `test1` is being changed to `apptest1`.

Note: You cannot rename an active boot environment.

You can run `beadm list` to verify that the name change has been made. In the example, you can see that the name has changed.

Destroying an Existing, Inactive Boot Environment with `beadm`

To destroy a boot environment, run `beadm destroy` *`beName`*.

```
# beadm destroy solaris-2
Are you sure you want to destroy solaris-2? This action cannot
be undone (y/[n]: y
# beadm list
BE          Active  Mountpoint Space   Policy  Created
--          ------  ---------- -----   ------  -------
apptest1     -        -         146.0K  static  2011-11-23 07:34
solaris      NR       /          4.87G  static  2011-11-19 22:14
solaris-1    -        -        121.39M  static  2011-11-23 06:59
#
```


If you no longer need a particular boot environment, or if you want to make more room on your system, you can delete it by using the `beadm destroy` command with the boot environment name. In this example, the inactive boot environment `solaris-2` is being destroyed. The system asks you for confirmation before destroying the boot environment.

Note: You can use the `-F` option to force the boot environment to be destroyed.

To verify that the boot environment has been removed, you can run `beadm list` again. The boot environment should not be listed.

Consider the following when you are planning to destroy a boot environment:

- You cannot destroy the boot environment that is currently booted.
- The `beadm destroy` command automatically removes the destroyed boot environment's entry from the x86 GRUB menu or the SPARC boot menu.

Activating an Existing, Inactive Boot Environment with `beadm`

To activate a boot environment, run `beadm activate` *`beName`*.

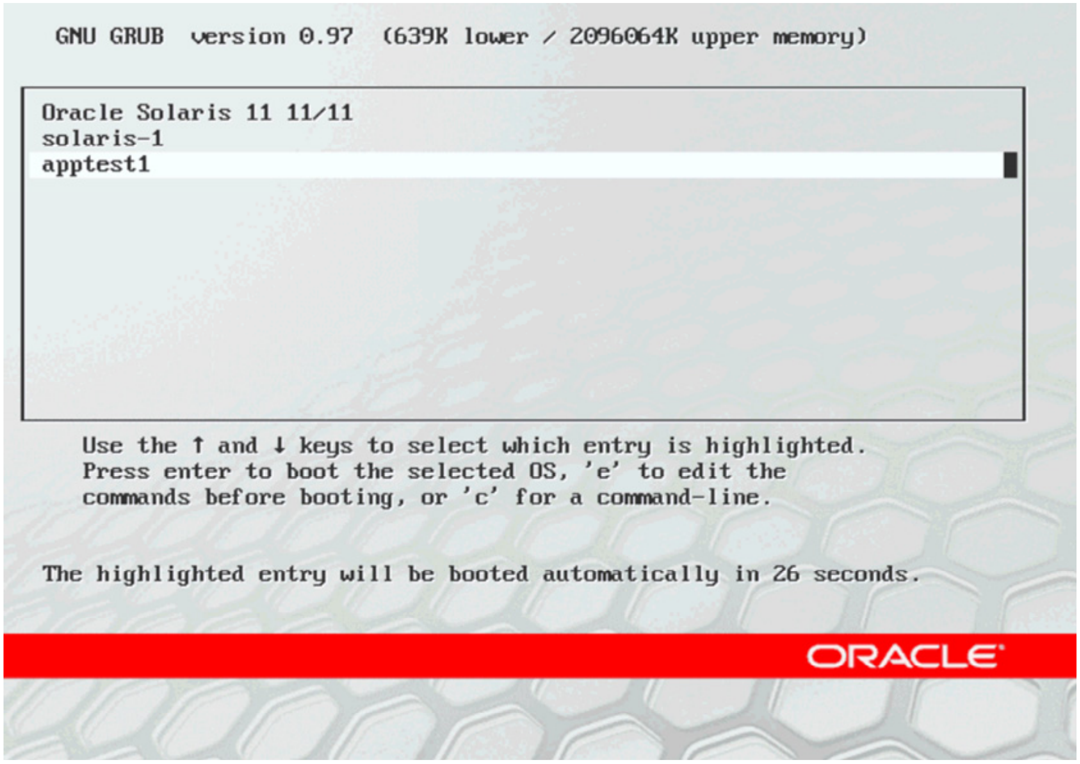
```
# beadm activate apptest1
# beadm list
BE         Active  Mountpoint Space   Policy  Created
--         ------  ---------- -----   ------  -------
apptest1    R       -         146.0K  static  2011-11-23 07:34
solaris     N       /          4.87G  static  2011-11-19 22:14
solaris-1   -       -        121.39M  static  2011-11-23 06:59
# init 6
```


To activate an existing, inactive boot environment, use the `beadm activate` command followed by the boot environment name. In this example, the `apptest1` environment is being activated.

To verify that the boot environment has been activated, you can run `beadm list` again. Notice that the Active status of the current boot environment `solaris` has changed from `NR` to `N`, and the newly activated boot environment `apptest1` now has an `R` in the Active column. Recall that `R` means that this boot environment will become the active boot environment on reboot.

To activate the new boot environment, you must reboot the system by using the `init 6` command.

Verifying the New Boot Environment



On an x86 machine, after the system has rebooted and the GNU GRUB menu appears, as shown in the example, you can verify that the new boot environment is now the default active boot environment. The default active boot environment is the highlighted entry.

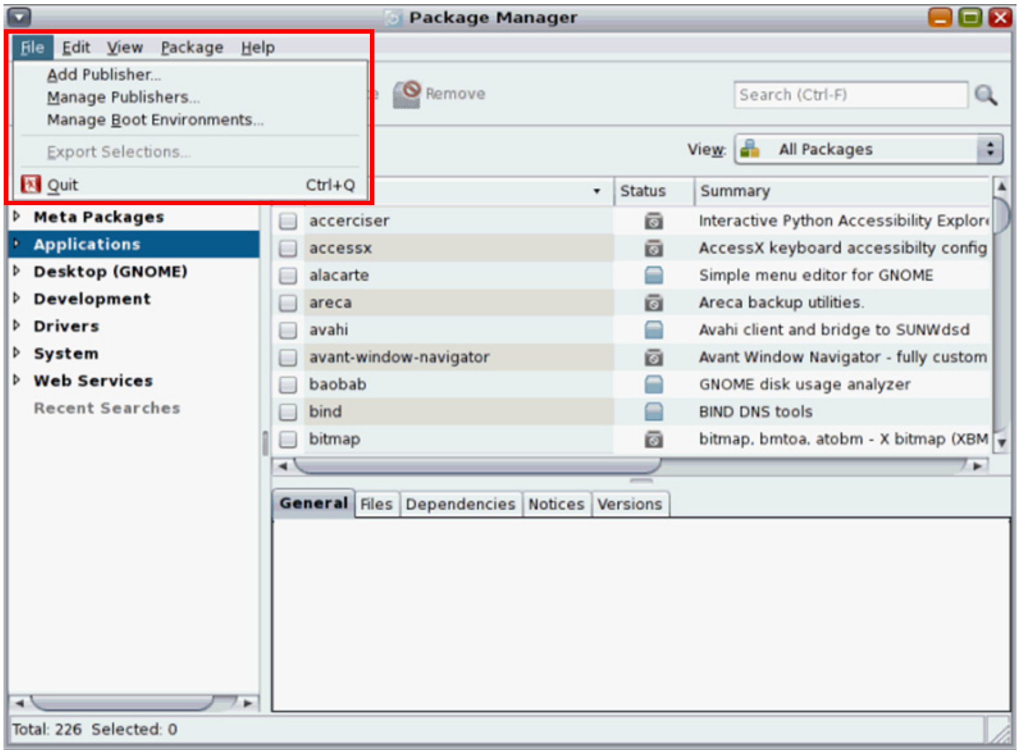
On a SPARC machine, when you have multiple BEs, you issue `init 0` to go to OBP (`ok` prompt) and then you use the following command sequence:

```
ok boot -L
Rebooting with command: boot -L
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0
File and args:      -L

1 Oracle Solaris 11 11.0
2 solaris-1
3 apptest1
Select environment to boot: [ 1 - 3 ]: 3
To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/apptest1

Program terminated
ok boot -Z rpool/ROOT/apptest1
```

Managing Boot Environments with Package Manager

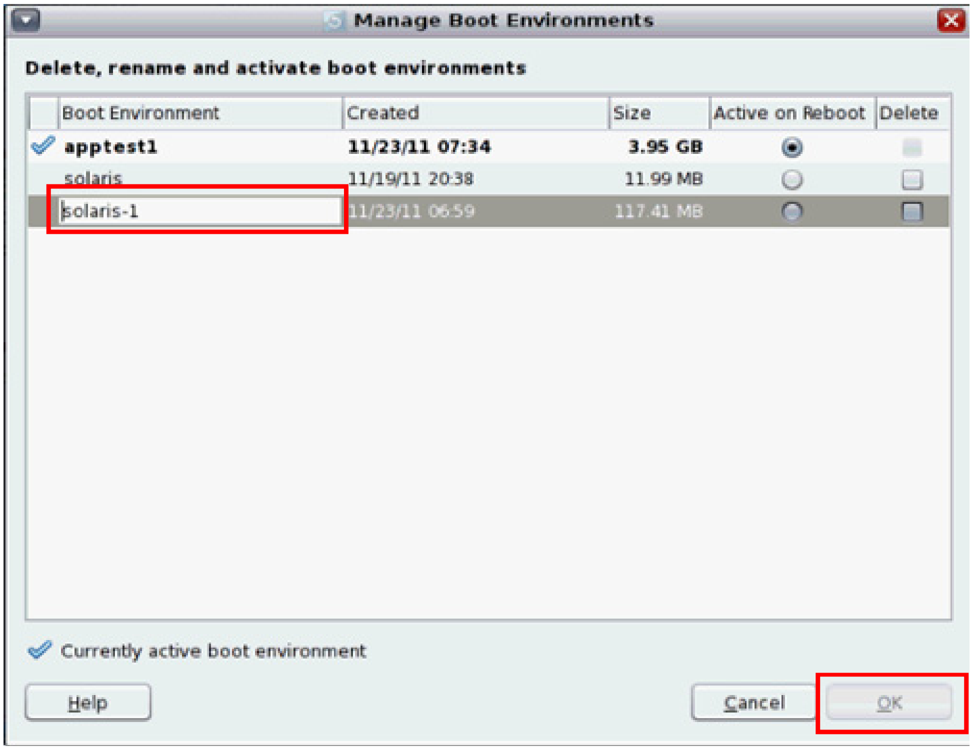


You now look at the boot environment administration tasks you can perform using Package Manager.

Although you cannot create new boot environments by using Package Manager, you can use this utility to rename, destroy, and activate existing, inactive boot environments that are on the system.

To manage boot environments with Package Manager, open Package Manager and then select Manage Boot Environments on the File tab. This launches the Manage Boot Environments window (shown in the next slide).

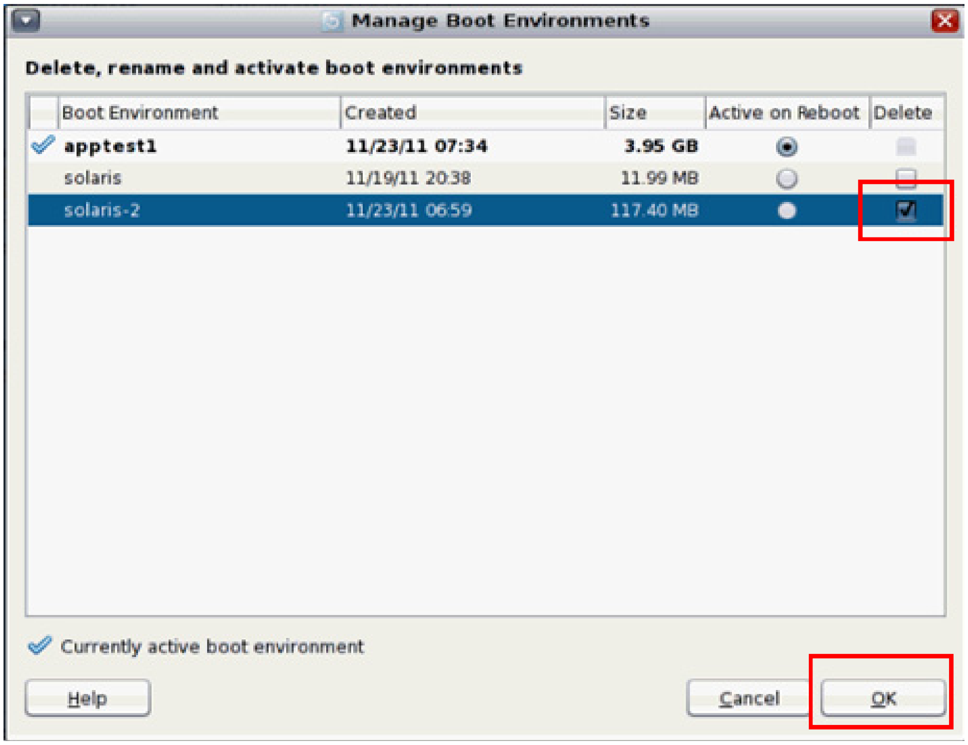
Renaming an Existing, Inactive Boot Environment with Package Manager



When the Manage Boot Environments window appears, you can rename an inactive boot environment by first double-clicking the name of the BE that you want to rename. Next, enter the new name, press Enter, and then click OK. In this example, you are renaming the `solaris-1` inactive boot environment to `solaris-2`.

When the Boot Environment Confirmation window appears, verify the change in the name, and then click OK.

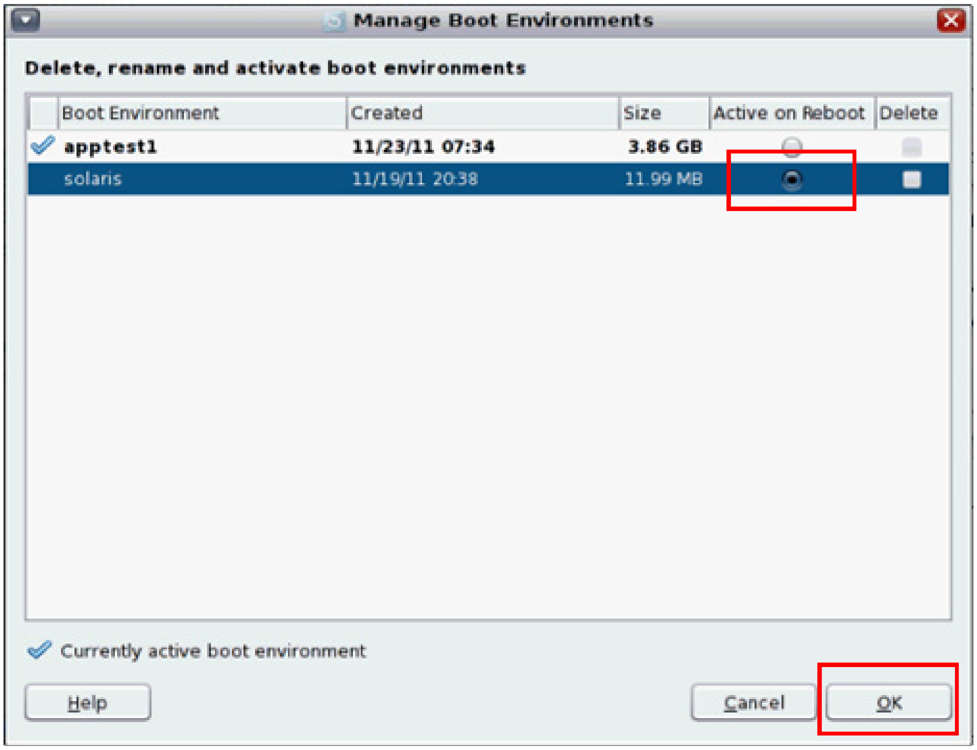
Deleting a Boot Environment with Package Manager



In Package Manager, you can destroy or delete an existing, inactive boot environment by selecting the associated check box in the Delete column for the boot environment you want to remove and then clicking OK. In this example, you are deleting the inactive boot environment `solaris-2`.

When the Boot Environment Confirmation window appears, click OK to confirm that you want to delete the boot environment.

Activating an Existing, Inactive Boot Environment with Package Manager



ORACLE

To activate an existing, inactive boot environment with Package Manager, select the "Active on Reboot" option for an inactive BE and then click OK. In this example, you have selected to activate the `solaris` boot environment on reboot.

To confirm your selection, click OK when the Boot Environment Confirmation window appears. The next time you reboot the system from the CLI, the new boot environment becomes the default active boot environment.

Practice 3-4 Overview: Administering Boot Environments

This practice covers the following topics:

- Listing the boot environments on the system
- Creating a new boot environment
- Activating an existing, inactive boot environment
- Rebooting the system

It should take you about 15 minutes to complete this practice.

Summary

In this lesson, you should have learned how to:

- Implement a plan for an Oracle Solaris 11 operating system software update
- Update the Oracle Solaris 11 operating system by using IPS
- Manage software packages by using the Oracle Solaris 11 operating system
- Administer boot environments

4

Administering Services

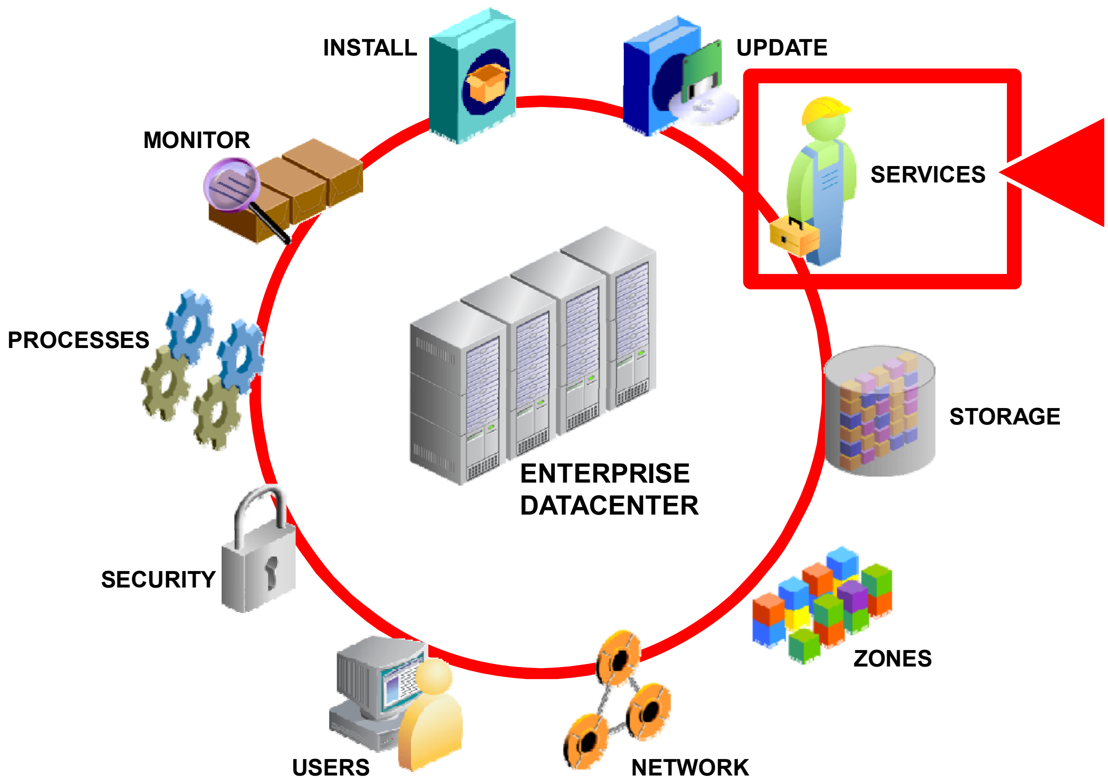
Objectives

After completing this lesson, you should be able to:

- Implement a plan for administering Oracle Solaris 11 services
- Administer SMF services
- Boot a system
- Shut down a system

In this lesson, you are introduced to the Service Management Facility (SMF) services and shown how to administer these services in accordance with a plan. The lesson concludes by discussing how to boot and shut down a system.

Workflow Orientation



Before you begin, take a moment to orient yourself to where you are in the job workflow. You have successfully installed the operating system and updated it. Now you are going to turn your attention to the services that are responsible for keeping the system running from day to day.

As a system administrator, it is your responsibility to manage these services and to ensure that the system and the business processes that are running on the system continue uninterrupted.

At the same time, if it is necessary to bring the system down for any reason, you need to know how to do that in a way that does not unexpectedly disrupt the services and resources that the system makes available to your end users.

Lesson Agenda

- Planning for Oracle Solaris 11 Services Administration
- Administering SMF Services
- Booting and Shutting Down a System

Planning for Oracle Solaris 11 Services Administration

Services Administration planning is required to ensure that:

- Default services run smoothly and efficiently
- System down time is planned
- The operating system boots and shuts down as expected

Your company's Oracle Solaris 11 implementation plan contains multiple subplans that include maintenance, emergency, and contingency planning. The testing your team is doing on the software's default functionality will provide information in support of these subplans. The focus of services administration planning is to ensure the following:

- The default services that are running on the system are running smoothly and efficiently.
- Any system down time is planned and, if not planned, at least minimized through effective emergency and contingency planning.
- The operating system boots and shuts down as expected.

In this section, you are introduced to the Oracle Solaris 11 services as well as the system boot and shutdown processes.

Service Management Facility

- Provides structure for managing:
  - System services
  - Interaction of services with other services
- Contains information about:
  - Procedures to start, stop, and restart services
  - Service startup behavior and status
  - Misconfigured services (such as an explanation of why a service is not running)
- Provides individual log files for each service

Oracle Solaris 11 uses a feature called the Service Management Facility (SMF) that provides a structure for managing system services and the interaction of services with other services. SMF contains information about how to start, stop, and restart services as well as information about a service's startup behavior and status. SMF also provides information about misconfigured services and explanations of why a particular service is not running. In addition, SMF provides log files for each service.

What Is a Service?

- An entity that provides a resource or list of capabilities to applications and other resources
- The software state of a device (for example, a configured network device or mounted file system)
- Is structured within SMF by:
  - Category (examples: `application`, `network`, `system`)
  - Service name (examples: `login`, `SSH server`, `hostid`)
  - Instance name: Specific configuration of a service (example: `default`)

Note: There can be multiple instances of a service.

A *service* is an entity that provides a resource or list of capabilities to applications and other services, both local and remote. A service can also be the software state of a device, such as a configured network device or a mounted file system.

Services that are managed by SMF are structured by category, service name, and instance name.

Examples of categories include:

- `application`
- `device`
- `legacy`
- `milestone`
- `network`
- `platform`
- `site`
- `system`

Examples of service names are `login`, `SSH server`, `and hostid`.

An instance is a specific configuration of a service. For example, a web server is a service. A specific web server daemon that is configured to listen on port 80 is an instance. Examples of an instance name include `default`.

Note: There can be multiple instances of a service. For example, a system can have more than one configured network interface or more than one mounted file system.

Identifying a Service Instance in SMF

Example FMRI:

`svc:/system/filesystem/root:default`

where:

- The prefix `svc` indicates that this service is managed by SMF
- The highest category of the service is pointing to the system facilities (`system`)
- Within `system` the next level category is `filesystem`
- The next lower category is `root`, which is pointing to the `root` file system.
- The service name is `system/filesystem/root:default`
- `default` is an instance of the service

SMF identifies each instance of a service by a service identifier. This service identifier is in the form of a Fault Management Resource Identifier or FMRI. (Recall that the Oracle Solaris 11 packages also use an FMRI.) The FMRI indicates the service type or category, service name, and instance name. An example of a service FMRI is shown in the slide.

Note: You do not always have to use the full FMRI when executing a command for a service. You can use a shorter form. For example, with `svc:/system/filesystem/root:default`, you can just use `filesystem/root`.

Service Configuration Repository

- Stores state and configuration information about each service instance
- Is named `/etc/svc/repository.db`
- Is managed by the `svc.configd` daemon

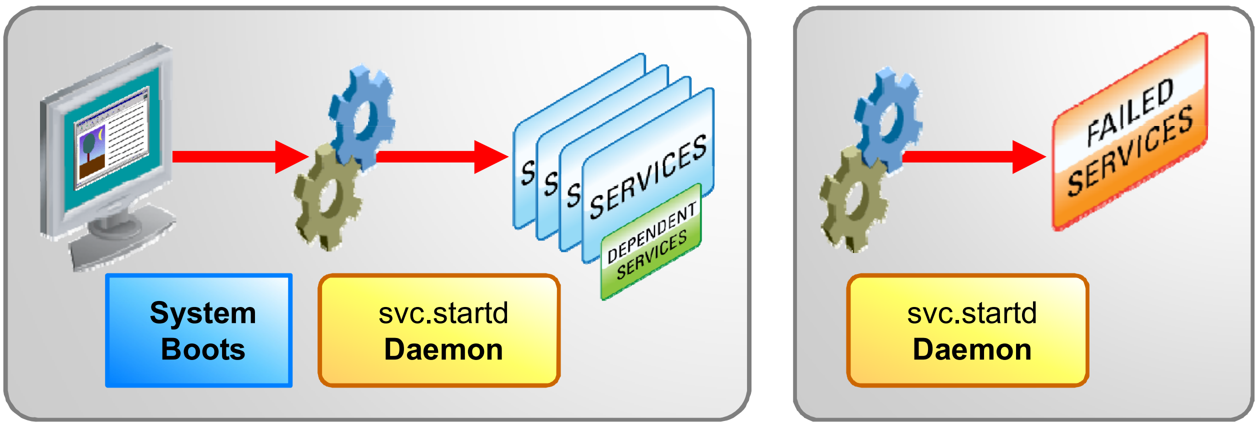
In SMF, state and configuration information about each service instance is stored in the service configuration repository. The repository is distributed among local memory and local disk-based files and is called `/etc/svc/repository.db`.

The repository is managed by the `svc.configd` daemon. This daemon is the interface between the repository and the user, and ensures that a consistent picture of the repository is presented to the user.

A service known as the manifest-import service takes a backup of the repository during reboot. This backup of the repository ensures that failback is possible.

SMF Master Restarter Daemon (`svc.startd`)

- Manages service dependencies for the entire system
- Makes sure the system boots properly
- Is responsible for starting services, restarting services, and shutting down services



SMF controls the starting and restarting of services through the master restarter daemon. The master restarter daemon (`svc.startd`) is responsible for managing service dependencies for the entire system. When a system is first started, it is the `svc.startd` daemon that ensures that the system boots properly and that the services are started appropriately in the right order.

Note: The `svc.startd` daemon is also responsible for ensuring that the system boots to the appropriate milestone. A milestone is a special service that depends on a set of services to be started before its start is satisfied. Services that depend on the milestone service will have their dependency satisfied and can start after other dependencies are satisfied. A milestone can be regarded as a system state to reach. If no milestone is specified at boot up, `svc.startd` boots to the built-in milestone `all`, which includes all the system-enabled services.

Currently, the milestones that can be used at boot time are the following:

- `none`
- `single-user`
- `multi-user`
- `multi-user-server`
- `all`

When services are first starting up, the `svc.startd` daemon retrieves service information from the configuration repository and then starts the services when their dependencies have been met. The master restarter daemon is also responsible for restarting failed services and for shutting down services whose dependencies are no longer satisfied.

Run Levels

Run Level	Resulting State	Description
`0`	Exit the OS.	The operating system is shut down, and it is safe to turn off power to the system.
`s` or `S`	Single-user state	A single user can log in. Some file systems are mounted and accessible.
`2`	Multiuser state	Multiple users can access the system and all file systems.
`3`	Multiuser level with server	All system resources are available, and multiple users can log in. This is the default run level.
`5`	Machine powers down.	The system shuts down and then powers off the machine.
`6`	Boot to multiuser level with server.	The system shuts down to run level `0` and then reboots to run level `3`.

SMF Milestones

Run Level	SMF Milestone FMRI
`S`	`milestone/single-user:default`
`2`	`milestone/multi-user:default`
`3`	`milestone/multi-user/server:default`

In addition, the `svcadm` command can be used with the `milestone` subcommand to change the run level of a system by selecting a milestone at which to run. The table presents the most commonly used milestones.

Note: The `svcadm` command enables you to manipulate service instances. You are shown how to use the `svcadm` command later in this lesson to administer services.

Oracle Solaris Boot Design

- SPARC and x86 boot design architectures are similar.
- Differences include how the boot device and arguments are selected at boot time.
- SPARC uses the open boot PROM (OBP) and its commands.
- x86 uses the BIOS and the GRUB menu.
- By default, SPARC and x86 platforms have one primary boot archive and one fail-safe boot archive.

Although Oracle Solaris 11 is designed to run continuously, there are times when the system needs to be rebooted or shut down, as you saw with the OS installation and update procedures. Having a basic understanding of the steps that the system goes through when it boots and shuts down is a critical skill for all system administrators to have.

The Oracle Solaris boot design architecture for SPARC is similar to the x86 architecture. The two architectures differ in how each selects the boot device and arguments at boot time. The SPARC platform uses the open boot PROM (OBP) and its commands for selections, while the x86 platform uses the BIOS and the GRUB menu. On the SPARC and x86 platforms, there is one primary boot archive and one fail-safe boot archive by default.

Note: PROM stands for “programmable read-only memory.” BIOS stands for Basic Input/Output System and GRUB stands for Grand Unified Bootloader.

Before you continue, familiarize yourself with the SPARC open boot PROM and x86 BIOS and GRUB functionalities.

Boot PROM for SPARC Systems

The Boot PROM firmware:

- Provides basic hardware testing and initialization before loading the operating system
- Enables booting from a wide range of devices
- Provides a user interface
- Has access to a standard set of device drivers

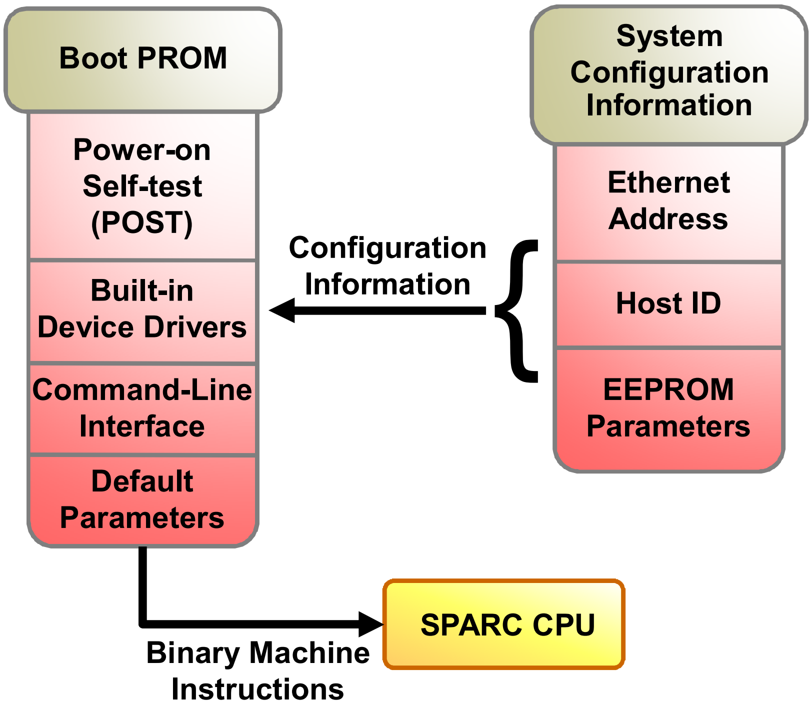
All Oracle Solaris SPARC systems have resident boot PROM firmware that controls the operation of the system before the operating system has been booted.

In addition to booting the operating system, the OpenBoot architecture enables the boot PROM firmware to do the following:

- Test and initialize system hardware.
- Determine the system's hardware configuration.
- Provide an interactive interface for configuration, testing, and debugging.
- Enable the use of third-party devices.

The boot PROM has access to a standard set of generic device drivers. The system needs these drivers to access and control the buses and the boot device to boot the system properly. All versions of the OpenBoot architecture allow a third-party board to identify itself and load its own plug-in device driver. Each device identifies its type and furnishes its plug-in device driver when requested by the OpenBoot PROM during the system hardware configuration phase of the boot process.

Boot PROM Initialization Sequence



When a system's power is turned on, a low-level power-on self-test (POST) is initiated. This low-level POST code is stored in the boot PROM and is designed to test the most basic functions of the system hardware.

After successful completion of the low-level POST phase, the boot PROM firmware takes control and performs the following initialization sequence:

- Probes the memory and then the CPU
- Probes bus devices, interprets their drivers, and builds a device tree
- Installs the console

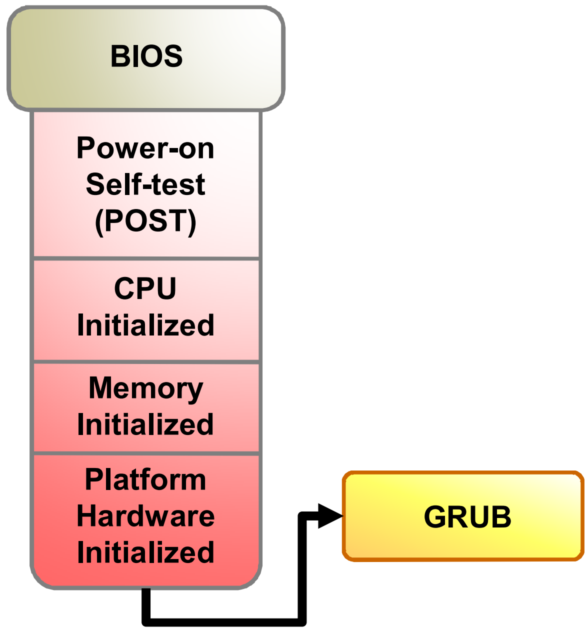
As part of this process, system configuration information is provided to the boot PROM. The system configuration information includes the following:

- Ethernet or MAC address, such as `8:0:20:5d:6f:9e`
- System host ID, such as `805d6f9e`
- User-configurable parameters that have been modified from the default settings. The user-configurable parameters are known as NVRAM variables or EEPROM parameters. They allow an administrator to control things such as the default boot device, the level of POST, and so on.

After the boot PROM initializes the system, the banner is displayed on the console. The system checks parameters stored in the boot PROM and NVRAM to determine whether and how to boot the operating system.

Note: When the boot process has completed and the operating system is running, you see a login prompt displayed on the console. When the operating system is not running, the `ok` prompt is displayed. You learn how to boot the system from the `ok` prompt later in this lesson.

BIOS and GRUB Initialization Sequence for x86 Systems



When an x86-based system is powered on, the system firmware in the BIOS executes a POST and initializes the CPU, the memory, and the platform hardware. When the initialization sequence has completed, the BIOS loads the boot loader from the configured boot device, and then passes control of the system to the boot loader. The boot loader is the GNU GRUB and is responsible for loading a boot archive into the system's memory. The boot archive contains the kernel modules and configuration files that are required to boot the system.

When the system has booted, the GRUB menu appears. This menu provides a list of boot entries from which to choose. A boot entry is an OS instance that is installed on your system. GRUB also has a command-line interface that is accessible from the menu interface for performing various boot commands. You have seen an example of this menu in the previous two lessons.

Boot Process

Boot Loader Phase	Root file system archive is loaded.
Booter Phase	Boot archive is read and executed.
Ramdisk Phase	Kernel image is extracted and executed.
Kernel Phase	Oracle Solaris is initialized and root file system is mounted.
init **Phase**	The `init` daemon starts the `svc.startd` daemon.
svc.startd **Phase**	The `svc.startd` daemon starts system processes.

Now that you have a better understanding of the boot PROM and BIOS/GRUB functionalities, you look at the actual boot process that is shared by both SPARC and x86 systems.

The boot process consists of six independent phases:

- **Boot loader phase:** During this phase, the Oracle Solaris root file system archive is loaded from the media to memory.
- **Booter phase:** During this phase, the boot archive is read and executed. Note that this is the only phase of the boot process that requires knowledge of the root file system format. The boot archive is a ramdisk image that contains all of the files that are required for booting a system. The SPARC boot archive is identical to an x86 boot archive.
- **Ramdisk phase:** During this phase, the ramdisk extracts the kernel image from the boot archive and then executes it. Neither the booter nor the kernel needs to know about the format of the boot archive.

- **Kernel phase:** During this phase, Oracle Solaris is initialized and a minimal root file system is mounted on the ramdisk that was constructed from the boot archive. In some environments, such as an installation, the ramdisk is used as the root (`/`) file system and remains mounted. The ramdisk contains a set of kernel files and drivers that is sufficient to mount the root file system on the specified root device. The kernel then extracts the remainder of the primary modules from the boot archive, initializes itself, mounts the real root file system, and then discards the boot archive. In addition, the kernel runs the `/sbin/init` program, which in turn starts the `init` daemon.
- **`init` phase:** During this phase, the init daemon initializes stream modules, sets up the system for a correct response to a power-fail shutdown, and starts the `svc.startd` daemon.
- **`svc.startd` phase:** During this phase, the `svc.startd` daemon starts the system processes.

How Oracle Solaris Boot Archives Are Managed

- Boot archive updates and verification are handled automatically by the `bootadm` command.
- During an installation or upgrade, an initial boot archive is created.
- During normal shutdown, the boot archive contents are compared with the root file system.
- If inconsistencies are found, the boot archive is rebuilt to make sure that the boot archive and root file system are synchronized.

System Shutdown Process

Oracle Solaris boot archive updates and verification on both the SPARC and x86 platform are handled automatically by the `bootadm` command. During an installation or upgrade, the `bootadm` command creates an initial boot archive. During the process of a normal system shutdown, the shutdown process compares the boot archive's contents with the root file system. If there are any inconsistencies, the system rebuilds the boot archive to ensure that, after reboot, the boot archive and root file system are synchronized.

Note: SMF manages the following boot archive services:

```
svc:/system/boot-archive:default
svc:/system/boot-archive-update:default
svc:/system/boot-config:default
```

Implementing the Services Administration Plan

Your assignment is to test:

- SMF services monitoring commands
- SMF services enable, disable, and restart commands
- System boot and shutdown functionality

Now that you have been introduced to the SMF services and have a better understanding of run levels and the boot process for both SPARC and x86 systems, it is time for you to help implement your part of the services administration plan.

As part of your company's preparations to implement Oracle Solaris 11 and as part of its services administration plan, you have been tasked with testing the SMF monitoring commands as well as the enable, disable, and restart commands. After you are done testing the services, you test to make sure that the system boots to the proper run level after you issue the associated shutdown commands.

In the next section, we describe the commands you use to perform these tasks.

Quiz

What is the service category in the `svc:/network/ssh:default` service FMRI?

a. `svc`

b. `network`

c. `ssh`

d. `default`

Answer: b

Quiz

Which of the following daemons is responsible for starting services?

a. `svc.startd`

b. `/etc/init`

c. `svc.configd`

Answer: a

Lesson Agenda

- Planning for Oracle Solaris 11 Services Administration
- Administering SMF Services
- Booting and Shutting Down a System

Administering SMF Services

- Monitoring services with `svcs`
  - Listing services information
  - Displaying the status of a service
  - Setting up service state transition notifications
  - Displaying the service dependents
  - Displaying the dependencies of a service
- Administering services with `svcadm`
  - Disabling a service
  - Enabling a service
  - Restarting a service

When a system is started, SMF starts up a number of services that continue to run in the background. When provided with the appropriate Service Management privileges, a system administrator can monitor, manage, and even configure the services.

As a new system administrator, your services administration tasks focus primarily on:

- Monitoring services by displaying service information by using the `svcs` command
- Performing basic administration (such as disabling and enabling services and restarting them) by using the `svcadm` command

In this section, you learn how to perform these tasks.

Note: More advanced services administration tasks, such as configuring and troubleshooting, are covered in the *Oracle Solaris 11 Advanced System Administration* course.

Listing Services Information

To list all the services currently running on the system, run `svcs`.

```
# svcs
STATE          STIME    FMRI
legacy_run     19:39:31 lrc:/etc/rc2_d/S89PRESERVE
online         19:38:20 svc:/system/svc/restarter:default
<output omitted>
```

To list all the services defined on the system, run `svcs -a`.

```
# svcs -a
STATE          STIME    FMRI
legacy_run     19:39:31 lrc:/etc/rc2_d/S89PRESERVE
disabled       19:38:25 svc:/network/physical:nwam
<output omitted>
```


To see what services are currently running on the system (including services that have been disabled temporarily), use the `svcs` command with no arguments, as shown in the first partial example. The `svcs` command displays information about service instances as recorded in the service configuration repository.

As you can see in the example, the `svcs` command output displays the state of the service, the time the service was started, and the service identifier (FMRI).

To see all the services that have been defined on the system, use the `svcs -a` command, as shown in the second partial example.

Displaying the Status of a Service Instance

To display the status of a service, run `svcs -l` *`FMRI`*.

```
# svcs -l svc:/network/ssh:default
fmri         svc:/network/ssh:default
name         SSH server
enabled      true
state        online
next_state   none
state_time   Mon Nov 28 07:39:31 2011
logfile      /var/svc/log/network-ssh:default.log
restarter    svc:/system/svc/restarter:default
contract_id  98
manifest     /etc/svc/profile/generic_limited_net.xml
manifest     /lib/svc/manifest/network/ssh.xml
dependency   require_all/none svc:/system/filesystem/local (online)
dependency   optional_all/none svc:/system/filesystem/autofs (online)
dependency   require_all/none svc:/network/loopback (online)
dependency   require_all/none svc:/network/physical:default (online)
dependency   require_all/none svc:/system/cryptosvc (online)
dependency   require_all/none svc:/system/utmp (online)
dependency   optional_all/error svc:/network/ipfilter:default (disabled)
dependency   require_all/restart file://localhost/etc/ssh/sshd_config (online)
```


To display the status information for an instance of a service, use the `svcs -l` command, followed by the service identifier, as shown in the example. The `-l` option, when used with the `svcs` command, displays all available information about the selected services and service instances, with one service attribute displayed for each line.

In this example, the status of the `ssh` service is displayed. The output displays:

- The FMRI for the service instance
- Whether the service is enabled or not
- The state of the service (in this case, the service is online)
- The next state of the service
- The service that is used to restart the service
- The contract ID
- Associated manifests
- A list of service dependencies

Note: SMF uses delegated restarters to restart services that share a set of common startup behaviors. In some cases, delegated restarters are used to provide more complex or application-specific restarting behavior. A delegated restarter's name is stored with the service, as shown in the example.

Identifying Service States

- `online`: Enabled and successfully started
- `offline`: Enabled but not yet running or available to run
- `disabled`: Not enabled and not running
- `legacy_run`: Running. The legacy service is not managed by SMF, but the service can be observed. This state is used by legacy services only.
- `uninitialized`: Starting up. This state is the initial state for all services before their configuration has been read.
- `maintenance`: Error encountered that requires administrative intervention
- `degraded`: Enabled but running at a limited capacity

A service instance can have different states. The list of service states is presented in this slide.

Setting Up Service State Transition Notifications

Monitored Transition States	
to-uninitialized	to-disabled
from-uninitialized	from-disabled
to-maintenance	to-online
from-maintenance	from-online
to-offline	to-degraded
from-offline	from-degraded

To set up the notifications:

1. Install the `smtp-notify` package.
2. Enable the notification service.
3. Configure the notifications.

SMF has a notification feature (also referred to as the notification framework) that notifies you through email messages of service state transitions and fault management events. The notification feature monitors the transition states presented in the table in the slide and uses a service called the Simple Mail Transfer Protocol (SMTP) service to send the email notification when a service changes states. This feature enables you to respond quickly to any changes in service states that might require immediate resolution.

Note: You can also configure the SMF notification feature to work with SNMP traps.

To use the notifications feature, you must install the `smtp-notify` package, enable the service that controls the notification feature, and then configure the notifications. You will now walk through each of these steps, beginning with the installation of the `smtp-notify` package.

Installing the `smtp-notify` Package

To install the SMF notification feature, run `pkg install service/fault-management/smtp-notify`.

```
# pkg install service/fault-management/smtp-notify
               Packages to install:     1
           Create boot environment:    No
               Services to restart:     1
DOWNLOAD                                PKGS       FILES    XFER (MB)
Completed                                1/1         4/4      0.0/0.0

PHASE                                      ACTIONS
Install Phase                                24/24

PHASE                                        ITEMS
Package State Update Phase                     1/1
Image State Update Phase                       2/2
```


Before you can use the SMF notification feature, you must first install it. The `smtp-notify` package that contains the notification functionality and the daemon that controls it is not installed by default. To install the package, you use the `pkg install` command followed by the package name (`service/fault-management/smtp-notify`), as shown in the example.

Enabling the `smtp-notify:default` Service

To enable the SMF notification service, run `svcadm enable svc:/system/fm/smtp-notify:default`.

```
# svcadm enable svc:/system/fm/smtp-notify:default
```

To confirm that the service is up and running, run `ps -ef | grep smtp-notify`.

```
# ps -ef | grep smtp-notify
    root  9818  9774   0 11:56:52 console     0:00 grep smtp-notify
noaccess  1060     1   0 11:45:9  ?           0:00 /usr/lib/fm/notify/smtp-notify
```


After you have installed the `smtp-notify` package, you need to enable the service that controls the notification feature's daemon. To do this, you use the `svcadm enable` command followed by `svc:/system/fm/smtp-notify:default`, as shown in the example.

To confirm the service is up and running, you can run the `ps -ef | grep smtp-notify` command, as shown in the example.

Note: You learn more about the `ps` command in the lesson titled "Managing System Processes and Scheduling System Tasks."

Configuring Service State Transition Notifications

To configure the service state transition notifications for all services, run `svccfg -s svc:/system/svc/global:default setnotify -g` *`service_transition_state`* `mailto:root@localhost`

```
# svccfg -s svc:/system/svc/global:default setnotify -g \
from-online mailto:root@localhost
```

To configure notifications for a single service, run `svccfg -s` *`FMRI`* `setnotify from-online mailto:root@localhost`

```
# svccfg -s svc:/network/http:apache22 setnotify \
from-online mailto:root@localhost
```


After you have verified that the `smtp-notify` service is up and running, the final step in the SMF notification setup process is to configure the notifications. You can configure notifications for all services or a single service. To receive notifications for all services, you run the command `svccfg -s svc:/system/svc/global:default setnotify -g` *`service_transition_state`* `mailto:root@localhost`. In the first example, you are setting up a notification to occur if any service state changes from the online state to any other state (`from-online`).

Note: The `svccfg` command is used to modify service configurations. This `svccfg` command is covered in more detail in the *Oracle Solaris 11 Advanced System Administration* course.

To configure notifications for a single service, you use the same command but you specify the FMRI for a specific service; you don't need to specify the `-g` option before identifying the service state change that you want to monitor. In the second example, you are setting up a notification to alert you if the `appache22` service state changes from online to any other state.

Note: You can specify multiple keywords for service transition states by separating them by a comma, as in the following example:

```
    # svccfg -s svc:/system/svc/global:default setnotify -g from-
online,to-online  \ mailto:root@localhost
```

Service State Transition Notification: Example

```
# mail
From noaccess@solaris.local Mon Nov 28 06:12:59 2010
Date: Mon, 28 Nov 2011 06:12:59 +0100 (CET)
From: No Access User
Message-Id: <201011180512.oAI5CxVV001278@solaris.local>
Subject: Fault Management Event: solaris:SMF-8000-YX
To: root@solaris.local
Content-Length: 668

SUNW-MSG-ID: SMF-8000-YX, TYPE: defect, VER: 1, SEVERITY: major
EVENT-TIME: Mon Nov 28 06:12:58 MST 2010
PLATFORM: VirtualBox, CSN: 0, HOSTNAME: solaris
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: 345ed233-8d58-ef72-d050-c552d0c78670
DESC: A service failed - a start, stop or refresh method failed.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/network/http:apache22 is unavailable.
REC-ACTION: Run 'svcs -xv svc:/network/http:apache22' to determine the
generic reason why the service failed, the location of any logfiles, and
a list of other services impacted. Please refer to the associated
reference document at http://sun.com/msg/SMF-8000-YX for the latest
service procedures and policies regarding this diagnosis

? <Press Enter to see the next message>
```


This slide presents an example of a service state transition notification that is sent to you by email. The notification consists of two messages. The first message (an example of which is shown on this slide) provides information about the event itself. The notification example shown on this slide is alerting you to a change in state for the `apache22` service. As you can see, the notification contains information about the severity level of the message (in this example, major), the time the event took place, the platform on which the event took place and the name of the host, the source, the event ID, a description of the event (“`A service failed - a start, stop or refresh method failed.`”) and a URL that contains more information about the fault, the auto-response to the event (“`The service has been placed into the maintenance state.`”), the impact of the event (“`svc:/network/http:apache22 is unavailable.`”), and the recommended action (“`Run 'svcs -xv svc:/network/http:apache22' to determine the generic reason why the service failed, the location of any logfiles, and a list of other services impacted.`”).

Service State Transition Notification: Example

```
<continued from previous slide>

From noaccess@localhost.mydomain.com Mon Nov 28 12:04:23 2011
Date: Mon, 28 Nov 2011 12:04:23 -0600 (MST)
From: No Access User <noaccess@s11-server1.mydomain.com>
Message-Id: <201110051804.p95I4No8009910@s11-server1.mydomain.com>
Subject: s11-server1: svc:/network/http:apache22 online->offline
To: root@s11-server1.mydomain.com
Content-Length: 233

HOSTNAME: s11-server1
TIMESTAMP: Mon Nov 28 12:04:23 2011
FMRI: svc:/network/http:apache22
FROM-STATE: online
TO-STATE: offline
DESCRIPTION: The indicated service has transitioned to the offline state
REASON: a restart was requested

? q
#
```


The second message is from the `smtp` service and provides the service's "from-state" and "to-state" transition information. In this example, you can see that the `apache22` service has gone from the online state to the offline state.

To exit `mail`, use the `q` command.

Managing Service State Transition Notifications

To view configured notifications, run `svccfg -s svc:/system/svc/global:default listnotify`.

```
# svccfg -s svc:/system/svc/global:default listnotify

    Event: from-online (source: svc:/system/svc/global:default)
        Notification Type: smtp
            Active: true
            to: root@localhost
```

To stop all notifications, run `svccfg -s svc:/system/svc/global:default delnotify -g all`.

```
# svccfg -s svc:/system/svc/global:default delnotify -g all
```


You can view the notifications that are configured for a system by using the `svccfg listnotify` subcommand, as shown in the first example. Here, you can see that you have configured a notification for all services to be sent to you if there is a service state change from online to any other state.

If you want to stop service state transition notifications, you can do so using the `delnotify` subcommand, as shown in the second example. The command shown here stops all notifications.

Displaying the Service Dependents

To display service dependents, run `svcs -D` *`FMRI`*.

```
# svcs -D svc:/network/ssh:default
STATE          STIME    FMRI
online         19:39:38 svc:/milestone/self-assembly-complete:default
online         19:39:38 svc:/milestone/multi-user-server:default
```


Sometimes, services depend on one another to operate properly. If a service has dependents and the service fails, the services that depend on that service are affected. Whenever an issue with a service occurs that requires administrative intervention (such as taking the service down for maintenance), one of the first actions that is taken is to see what dependents that service has.

To determine which service instances depend on another service, use the `svcs -D` command followed by the service identifier, as shown in the example.

Displaying the Dependencies of a Service

To display service dependencies, run `svcs -d FMRI`.

```
# svcs -d svc:/network/ssh:default
STATE          STIME    FMRI
disabled       19:38:27 svc:/network/ipfilter:default
online         19:38:46 svc:/system/cryptosvc:default
online         19:38:38 svc:/network/loopback:default
online         19:39:26 svc:/system/utmp:default
online         19:38:44 svc:/network/physical:default
online         19:39:07 svc:/system/filesystem/local:default
online         19:39:24 svc:/system/filesystem/autofs:default
```


To understand why a particular service is not running, it can be helpful to know the dependencies that the service has.

To determine the services on which a service instance depends, use the `svcs -d` command followed by the service identifier, as shown in the example.

Disabling a Service

1. Use `svcs -D` *`FMRI`* to check the dependents of the service you want to disable.
2. Use `svcadm disable` *`FMRI`* to disable the service:

```
# svcadm disable svc:/network/ssh:default
```

3. Use `svcs -l` *`FMRI`* to verify that the service has been disabled:

```
# svcs -l svc:/network/ssh:default
fmri         svc:/network/ssh:default
name         SSH server
enabled      false
state        disabled
<output omitted>
```


The only reason to specifically disable and then enable a service is if changes need to be made before the service is enabled and after the service is disabled.

To stop a running service and prevent it from restarting, you must disable it. To disable a service, perform the steps outlined in the slide.

Notes for step 1: If this service has dependents that you need, you cannot disable the service.

Notes for step 2: In this example, you are disabling the SSH server.

Notes for step 3: The enabled status has changed from true to false, indicating that the service is currently disabled.

When you have disabled the service, the service status change is recorded in the service configuration repository. The disabled state will persist across reboots. This means that the only way to get the service running again is to enable it.

Note: You can also temporarily disable a service with the `-t` option. With this option, the service returns to an online state on reboot.

Enabling a Service

1. Use `svcs -l` *`FMRI`*`|grep online` to determine whether service dependencies are satisfied.
2. Use `svcadm enable` *`FMRI`* to enable the service:

```
# svcadm enable svc:/network/ssh:default
```

3. Use `svcs` *`FMRI`* to verify that the service has been enabled:

```
# svcs svc:/network/ssh:default
STATE          STIME    FMRI
online         20:17:40 svc:/network/ssh:default
```


To enable a service, perform the steps outlined in the slide.

Notes for step 1: If the service is online, the service dependencies are satisfied. If the service is not online, use `svcadm enable -r` *`FMRI`* to recursively enable all dependencies.

Notes for step 2: In this example, we are enabling the SSH server.

Notes for step 3: The state should read `online`.

Again, the service status change is recorded in the service configuration repository. The enabled state persists across system reboots as long as the service dependencies are met.

Refreshing and Restarting a Service

To refresh a service, run `svcadm refresh` *`FMRI`*.

```
# svcadm refresh svc:/network/ssh:default
```

To restart a service, run `svcadm restart` *`FMRI`*.

```
# svcadm restart svc:/network/ssh:default
```


If a service is currently running but needs to be restarted due to a configuration change or some other reason, the service can be restarted without your having to enter separate commands to stop and start the service. Some configuration changes require only a refresh, while others require a restart.

To refresh a service, use the `svcsadm refresh` command followed by the service identifier, as shown in the example. In this example, you are refreshing the `ssh` service.

To restart a service, use the `svcsadm restart` command followed by the service identifier, as shown in the example. In this example, you are restarting the `ssh` service.

Note: When a refresh is done, the running snapshot is taken/updated. After this, properties can be queried from that snapshot to get a consistent picture. For example, if a service needs two properties to determine behavior, those two properties can be set individually and then refreshed into the service's running environment.

Practice 4-1 and Practice 4-2 Overview: Administering Services and Administering SMF Notifications

These practices cover the following topics:

- Enabling and disabling services
- Displaying the services
- Exploring the service dependencies
- Administering the SMF notification feature

In the practices for Lesson 4, you are presented with three tasks designed to reinforce the concepts presented in the lecture portion of this lesson. You will have the chance to perform the following tasks:

- **Practice 4-1:** Administering services
- **Practice 4-2:** Administering SMF notifications
- **Practice 4-3:** Booting and shutting down the system

You will find Practices 4-1 and 4-2 in your *Activity Guide*. Each practice should take you about 25 minutes to complete.

Lesson Agenda

- Planning for Oracle Solaris 11 Services Administration
- Administering SMF Services
- Booting and Shutting Down a System

Booting a System

- Booting a SPARC-based system
  - To run level `3` (multiuser-server milestone)
  - To run level `S` (single-user milestone)
- Booting an x86-based system
  - To run level `3` (multiuser-server milestone)
  - To run level `S` (single-user milestone)
- Initiating a fast reboot

The steps for booting a system vary depending on the type of system (SPARC or x86) and the run level or milestone you want the system to achieve. In this section, you learn how to boot both SPARC and x86 systems to run level `3` (multiuser-server milestone) and level `S` (single-user milestone). You also take a brief look at how to initiate a fast reboot of both SPARC and x86 systems.

Note: There are many reasons for needing to boot a system. The methods for booting depend on what task you are trying to accomplish. This course focuses on just two ways to boot the system. For information about other reasons and ways to boot a SPARC or x86 system, see the *Oracle Solaris Administration: Common Tasks* guide or the `boot(1M)` man page.

Booting a SPARC-Based System

- Booting a SPARC system to run level `3` (multiuser-server milestone)
- Booting a SPARC system to run level `S` (single-user milestone)

You first learn about booting a SPARC system to run level `3` and then to run level `S`. You also examine when and why you perform these tasks.

Booting a SPARC System to Run Level 3 (Multiuser-Server Milestone)

1. Boot the system to run level 3 by using the `boot` command at the `OK` prompt.

```
ok boot
Resetting ...
<output omitted>
```

2. When prompted, log in to the system.
3. Verify that the system has booted to run level 3.

```
# who -r
. run-level 3 Nov 28 11:32 3 0 S
#
```

ORACLE

After performing a system hardware maintenance task or shutting down the system, you might be ready once again to make all system resources available to multiple users. To achieve this system state, you must boot the operating system to run level 3 (multiuser-server milestone). The assumption is that the operating system is at run level 0.

To boot a system that is currently at run level 0 to run level 3, perform the steps shown in the slide.

Note for step 1: The automatic boot procedure displays a series of startup messages and brings the system to run level 3.

Note for step 2: The login prompt is displayed during the boot process.

Booting a SPARC System to Run Level S (Single-User Milestone)

- Boot the system to run level S by using the `boot -s` command at the OK prompt.

```
ok boot -s
```

- When prompted, enter the root password.

```
SINGLE USER MODE
Root password for system maintenance (control-d to bypass):
xxxxxx
```

- Verify that the system is at run level S.

```
# who -r
. run-level S Nov 29 10:15 S 0 S
#
```


To boot a system that is currently at run level `0` to run level `S`, perform the steps shown in the slide. Single-user milestone is used for system maintenance tasks.

Note for step 3: At this point in the procedure, you can perform the system maintenance task that required you to change the system to run level `S`. When you are done with your maintenance task, you must press Ctrl + D to bring the system to the multiuser state (run level `3`, which is the default state).

Booting an x86 System

- Booting an x86 system to run level `3` (multiuser-server milestone)
- Booting an x86 system to run level `S` (single-user milestone)

You now learn about how to boot an x86 system to run level `3` and then to run level `S`. You also look at when and why you perform these tasks.

Booting an x86 System to Run Level 3 (Multiuser-Server Milestone)

1. Reboot the system by using the `init 6` command.

```
# init 6
```

2. When the GRUB menu appears, press Enter to boot the default OS instance to run level `3`.
3. When the login prompt appears, log in to the system as `root`.
4. Verify that the system has booted to run level `3`.

```
# who -r
. run-level 3 Nov 28 11:32 3 0 S
#
```


To boot a system that is currently at run level `0` to run level `3` (multiuser-server milestone), perform the steps shown in the slide.

Note for step 1: If the system displays the `Press any key to reboot` prompt, press any key to reboot the system. You can also click the Reset button at this prompt. If the system is shut down, turn the system on with the power switch. When the boot sequence begins, the GRUB menu is displayed.

Note for step 2: If you do not choose an entry within 10 seconds, the system automatically boots to run level `3`. The login prompt is displayed when the boot process has finished successfully.

Booting an x86 System to Run Level `S` (Single-User Milestone)

1. Reboot the system by using the `init 6` command.
2. When the GRUB menu appears, enter `e` to edit the GRUB menu.
3. Use the arrow keys to choose the `kernel $ line`.
4. Enter `e` again to edit the boot entry.
5. To boot the system in single-user mode, type `-s` at the end of the boot entry line. Then press Return to go back to the previous screen.
6. To continue to boot the system in single-user mode, enter `b`.
7. When prompted, enter the `root` password.
8. Verify that the system is at run level `S`.

ORACLE

To boot a system that is currently at run level `0` to run level `S`, perform the steps shown in the slide. Single-user milestone is used for performing system maintenance.

Note for step 1: If the system displays the `Press any key to reboot` prompt, press any key to reboot the system. You can also click the Reset button at this prompt. If the system is shut down, turn the system on with the power switch. When the boot sequence begins, the GRUB menu is displayed.

Note for step 3: Depending on your system, you might not be able to use the arrow keys. If this is the case, use the ^ key to scroll up and the v key to scroll down.

Note for step 4: The GRUB edit menu is displayed, enabling you to add options and arguments to the `kernel $` line.

Note for step 5: You can specify other boot behaviors by replacing the `-s` option with other appropriate boot options. For a list of these options, see the *Oracle Solaris Administration: Common Tasks* guide or the `boot`(1M) man page.

Note for step 8: At this point in the procedure, you can perform the system maintenance task that required you to change the system to run level `S`. When you are done with your maintenance task, you must reboot the system.

Fast Reboot

- Bypasses firmware and boot loader processes to provide an extremely fast reboot
- Is controlled by the SMF
- Is implemented through a boot configuration service (`svc:/system/boot-config`) based on the setting of the `fastreboot_default` property:
  - Set to `true` on x86 systems by default
  - Set to `false` on SPARC systems by default

The Fast Reboot feature enables you to reboot a system, bypassing the firmware and boot loader steps. Fast Reboot implements an in-kernel boot loader that loads the kernel into memory and then switches to that kernel, so that the reboot process occurs within seconds.

The Fast Reboot feature is controlled by the SMF and implemented through a boot configuration service, `svc:/system/boot-config`. The `boot-config` service provides a way to set or change the default boot configuration parameters.

The `fastreboot_default` property of the `boot-config` service enables an automatic fast reboot of the system when either the `reboot` or the `init 6` command is used. When the `config/fastreboot_default` property is set to `true`, the system automatically performs a fast reboot without the need to use the `reboot -f` command. By default, this property's value is set to `false` on the SPARC platform and to `true` on the x86 platform. On the x86 platform, an additional property, `config/fastreboot_onpanic`, is implemented by default. This property enables a fast reboot of an x86 system in the event of a system panic.

Note: If you have a SPARC-based or x86-based system that has Fast Reboot enabled by default, you can perform a slow reboot by using the `reboot -p` command without having to reconfigure the `boot-config` service to disable the feature.

Initiating a Fast Reboot of a SPARC-Based System

To initiate a fast reboot of a SPARC system, run `reboot -f`.

```
# reboot -f
```


Note

You must assume the `root` role to perform this operation.

Initiating a Fast Reboot of an x86-Based System

Because Fast Reboot is the default boot mode on an x86 system, you can use either the reboot command with the `-f` option or the `init 6` command to initiate the reboot.

```
# reboot -f
```

```
# init 6
```


Note

You must assume the `root` role to perform this operation. The commands must be executed from a terminal window.

Shutting Down a System

- Shutting down a server:
  - The `shutdown` command is used.
  - Clean shutdown is performed.
  - Superuser privileges are required.
- Shutting down a stand-alone system:
  - The `init` command is used.
  - Clean shutdown is performed.
  - Superuser privileges are required.

As with booting, there are many reasons why it might be necessary to shut down a system. For example, some system administration tasks and emergency situations require that the system be shut down to a level where it is safe to remove power. Sometimes, a system needs to be brought to an intermediate level, where not all system services are available.

Unlike booting, the procedures for shutting down a system are very much the same for both SPARC-based and x86-based systems.

In this course, you learn how to perform a clean shutdown of the system by using the `shutdown` command for a server where you have multiple users, and by using the `init` command for a stand-alone system. The use of these commands ensures that all system services, processes, and the operating system are terminated normally. You need superuser privileges to execute either of these commands.

For information about other reasons and ways to shut down a system, see the *Oracle Solaris Administration: Common Tasks* guide.

Determining Who Is Logged In to a System

To determine who is logged in to a system, run `who`.

```
$ who
holly    console  Nov 28 07:30
kryten   pts/0    Nov 28 07:35 (starlite)
lister   pts/1    Nov 28 07:40 (bluemidget)
```

Best practice: Always send an additional email notification to users indicating that the server is going to be down for a specified amount of time.

When shutting down a system, your first task is to determine who is logged in to the system. The purpose of this task is to notify those users that the system is going to be shut down.

To determine which users are logged in to the system, use the `who` command, as shown in this example.

The data displayed in the first column identifies the username of the logged-in user. The data in the second column identifies the terminal line of the logged-in user. The data in the third column identifies the date and time that the user logged in. The data in the forth column, if present, identifies the host name if a user is logged in from a remote system. In this example, there are two remote users: `kryten` and `lister`.

Although the system automatically notifies users that the system is shutting down, it is a best practice to send an additional email notification to users to let them know that the server is going to be down for a specified amount of time. By doing so, you enable users to prepare for the system down time.

Shutting Down a Server

1. Determine who is logged in to the system.
2. Shut down the system by using the `shutdown` command with the `-i`*`init-level`*, `-g` *`grace-period`*, and `-y` options.
3. When prompted, enter the superuser password.
4. Verify that the system is at the run level you specified.
5. When you have completed your administration tasks, press Ctrl + D to return to the default system run level.
6. Verify that the system is at the run level that you specified in the `shutdown` command.

Specified Run Level	SPARC System Prompt	x86 System Prompt
0 (exit the OS)	`ok >`	`Press any key to reboot`
s or S (single-user milestone)	#	#
3 (multiuser -server milestone)	*`hostname`* `console login:`	*`hostname`* `console login:`

ORACLE

When shutting down a server that is operating at run level `3` (multiuser-server milestone), it is recommended that you use the `shutdown` command. This is primarily because users are notified of the impending shutdown, as are any systems that mount resources from that server. The system is brought to run level `S` by default. To perform a clean shutdown of a server, perform the steps shown in the slide.

Note for step 2: The `-i init-level` option brings the system to an `init` level that is different from the default of `S`. The choices are `0`, `1`, `2`, `5`, and `6`. Run levels `0` and `5` are reserved states for shutting the system down. Run level `5` goes a step further and powers down the machine. Run level `6` reboots the system. Run level `2` is available as a multi-user operating state.

The `-g grace-period` option indicates the time (in seconds) before the system is shut down. The default is 60 seconds. The `-y` option continues to shut down the system without intervention. If you do not specify the `-y` option, you are prompted to continue the shutdown process when there are 30 seconds remaining. For more information, see the `shutdown`(1M) man page.

When the `shutdown` command is initiated, the system sends out a warning. This warning is followed by a final shutdown message that the server is being shut down, which is sent to any users who are logged in and to any systems that mount resources from the server.

Note for step 4: After you have verified that the system is at the specified run level, you can perform the system administration tasks that required you to change the system's run level.

Note for step 6: The table in the slide shows the prompts that you see (by system type) based on the run level you specified in the `shutdown` command. For example, if you specified a run level of `0` (exit the OS), you should see the `ok >` prompt on a SPARC system and the `Press any key to reboot` prompt on an x86 system.

Shutting Down a Stand-Alone System

To bring a stand-alone system to run level 0, run init 0.

```
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
<output omitted>
```

To bring a stand-alone system to run level S, run init S.

```
# init s
#
INIT: New run level: S
The system is coming down for administration.  Please wait.
<output omitted>
```


When you need to shut down a stand-alone server, it is recommended that you use the `init` command. This is because the `init` command provides a faster system shutdown because, unlike with the `shutdown` command, users are not notified of the impending shutdown.

To bring a stand-alone system to run level `0` (where it is safe to turn off power), run the `init 0` command, as shown in the first example. When the system has completed its shutdown, you see the `ok` or `>` prompt on a SPARC system and the `Press any key to reboot` prompt on an x86 system.

To bring a stand-alone system to run level `S` (so that you can perform administration tasks), run the `init S` command, as shown in the second example. After the system has entered the maintenance mode, you see the `#` prompt on both SPARC and x86 systems.

Practice 4-3: Booting and Shutting Down the System

This practice covers the following topics:

- Booting an X86/64 host
- Shutting down an X86/64 host

This practice should take you about 25 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for administering Oracle Solaris 11 services
- Administer SMF services
- Boot a system
- Shut down a system

In this lesson, you were introduced to the SMF services and learned how to monitor, enable, disable, and restart them. You also learned how to boot and shut down a system cleanly.

5

Setting Up and Administering Data Storage

Objectives

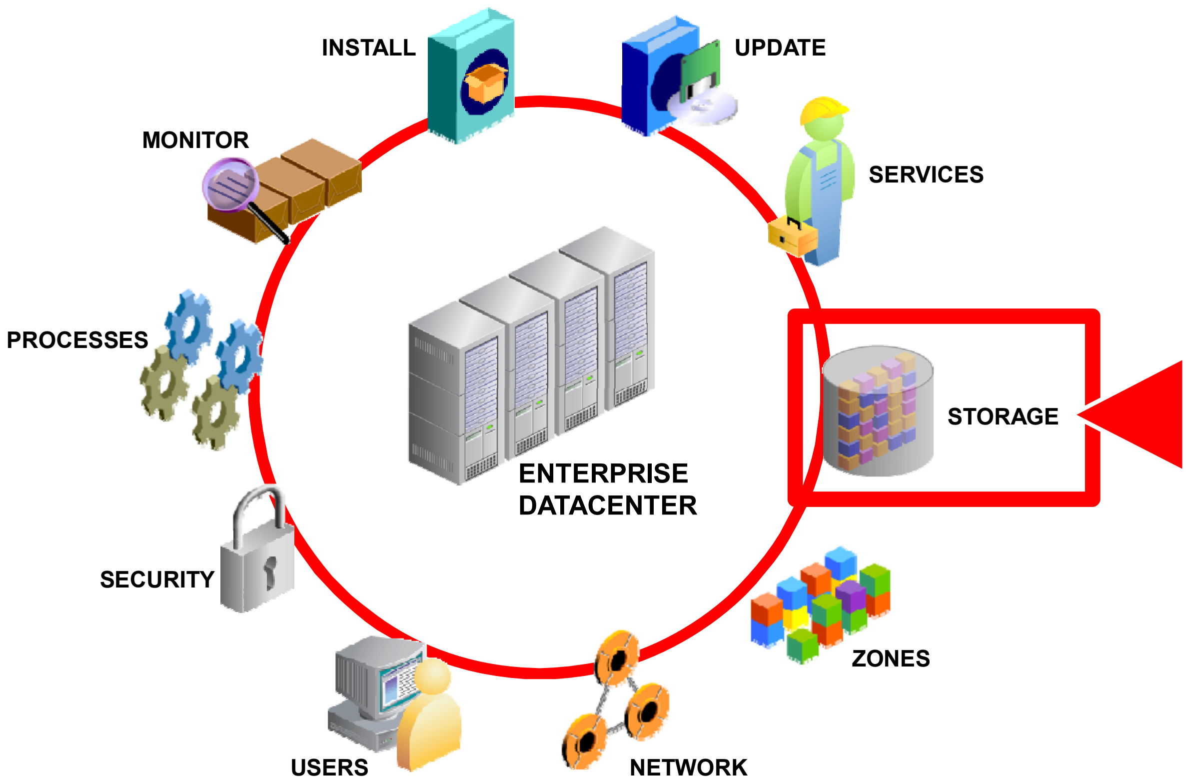
After completing this lesson, you should be able to:

- Implement a plan for data storage management
- Administer ZFS storage pools
- Administer ZFS file systems
- Administer ZFS snapshots and clones

Oracle Solaris 11 uses a very powerful and flexible technology known as ZFS to manage data storage. In this lesson, you learn how to set up and administer data storage by using this technology.

You are introduced to a plan for setting up storage on a local system. You then learn how to implement this plan to administer ZFS storage pools, file systems, snapshots, and clones.

Workflow Orientation



Before you begin, take a moment to orient yourself to where you are in the job workflow. You have successfully installed the operating system, updated it, and tested the SMF services as well as the system's boot and shutdown functionality. Now you are going to focus on your company's data storage needs.

In an enterprise environment, a system administrator is expected to manage the storage required for business applications data, such as customer information and product information. As a system administrator, you need to know how to configure the data storage environment based on your company's data storage requirements as well as how to administer the environment after it is in place.

Lesson Agenda

- Planning for Data Storage Management
- Administering ZFS Storage Pools
- Administering ZFS File Systems
- Administering ZFS Snapshots and Clones

Planning for Data Storage Management

Data Storage Management planning is required to ensure that:

- The appropriate type of storage pool configuration is selected that supports data redundancy and growth
- Data can be accessed, backed up, and restored quickly and easily

Because the proper storage and management of business application data is extremely important to your company's continued success, your company relies on Oracle Solaris 11 dynamic storage capabilities. The company has put together a comprehensive Data Storage Management plan to ensure that the data storage environment that is created can support the growing needs of the business and that the data that is stored is easily accessed, backed up, and quickly recoverable in the event of data loss.

In this section, you are introduced to one of Oracle Solaris's most powerful storage solution technologies, ZFS, and you learn how to use some of its primary features to support your company's data storage needs.

What Is ZFS?

- Two products in one:
  - Volume manager
  - File system
- Most scalable file system ever
  - 128-bit file system
  - Up to 256 trillion directory entries allowed
  - No limit to number of file systems or number of files contained within a file system
- Transactional file system
  - File system state is always consistent on disk.
  - Data is never overwritten.
  - The file system can never be corrupted through accidental loss of power or a system crash.

One goal of the ZFS design was to reduce the number of commands needed to create a usable file system. There are two primary commands to remember: the `zpool` command to manage pools and the `zfs` command to manage file systems.

ZFS has been designed from the ground up to be the most scalable file system ever. The file system itself is 128-bit, allowing for 256 quadrillion zettabytes of storage. Directories can have up to 256 trillion entries, and no limit exists on the number of file systems or the number of files that can be contained within a file system.

Moreover, ZFS is a transactional file system, which means that the file system state is always consistent on disk. Data is never overwritten, and any sequence of operations is either entirely committed or entirely ignored. This mechanism means that the file system can never be corrupted through accidental loss of power or a system crash. Although the most recently written pieces of data might be lost, the file system itself will always be consistent. In addition, synchronous data is always guaranteed to be written before returning, so it is never lost.

ZFS Storage Pools

- Storage pools are used to manage physical storage.
- No volume manager is required.
- Devices are aggregated into a storage pool.
- The storage pool:
  - Describes the physical characteristics of the storage
  - Acts as an arbitrary data store

ZFS uses the concept of storage pools to manage physical storage, thereby eliminating the need for a volume manager.

Instead of forcing you to create virtualized volumes, ZFS aggregates devices into a storage pool. The storage pool describes the physical characteristics of the storage (device layout, data redundancy, and so on), and acts as an arbitrary data store from which file systems and volumes can be created.

ZFS File Systems

- Share space with all file systems in the pool
- Grow automatically within the space allocated to the storage pool
- Immediately use additional space when new storage is added

File systems are no longer constrained to individual devices; this means they can share space with all file systems in the pool. You do not have to predetermine the size of a file system because file systems grow automatically within the space allocated to the storage pool. When new storage is added, all file systems within the pool can immediately use the additional space.

ZFS Snapshots

- Are a read-only view of a file system or volume
- Can be created quickly and easily
- Unlimited number allowed
- Consume no additional space initially
- Consume space as data within the active dataset changes
- Prevent data from being freed back to the pool
- When snapshots are destroyed, consumed space is released.

To protect the data that is stored in the ZFS storage pools and on the ZFS file systems, ZFS has snapshot and clone capabilities. A snapshot is a read-only view of a file system or volume. Snapshots can be created quickly and easily. In fact, ZFS allows you to take 2^{64} instantaneous snapshots of file systems. Initially, snapshots consume no additional space within the pool because they provide pointers to the data blocks. As data within the active dataset changes, the snapshot consumes space by continuing to reference the old data. As a result, the snapshot prevents the data from being freed back to the pool. When the snapshot is destroyed, the consumed space is released.

ZFS Clones

- Writable volume or file system
- Created from a snapshot
- Nearly instantaneous creation
- Initially consumes no additional disk space

A clone is a writable volume or file system created from a snapshot. A clone can be used to modify production data. As with snapshots, creating a clone is nearly instantaneous and initially consumes no additional disk space.

Now that you are more familiar with ZFS and some of its primary features, you return to the data storage planning activity.

Determining Your ZFS Storage Pool Requirements

As part of planning, you should identify your storage pool requirements:

- Devices
  - Disks of at least 128 MB in size
  - Not in use by other parts of the operating system
  - Entire disks formatted as a single, large slice or individual slices on a preformatted disk
- Level of data redundancy option
  - Non-redundant (striped) configurations
  - Mirrored
  - RAID-Z

To set up your ZFS environment, you must first create a ZFS storage pool based on the storage requirements that are identified during planning. The pool describes the physical characteristics of the storage and must be created before any file systems are created.

As part of planning, the devices that will store the business application data must be identified. The devices that are selected must meet the following criteria. They must be disks of at least 128 MB in size, and they must not be in use by other parts of the operating system. The devices can be entire disks that ZFS formats as a single large slice—which is the recommended approach—or they can be individual slices on a preformatted disk.

The level of redundancy is another storage requirement that should be addressed during planning. ZFS supports multiple types of data redundancy. The level of redundancy determines what types of hardware failures the pool can withstand. ZFS supports non-redundant (striped) configurations as well as mirroring and RAID-Z (a variation on RAID-5).

ZFS Storage Pool Components

The following components can be used in a ZFS storage pool:

- Disks
- Slices
- Files
- Virtual devices

ZFS Storage Pool Components: Disks

- Any block device of at least 128 MB in size
- Typically, a hard drive visible to the system in the `/dev/dsk` directory
- Whole disk (`c1t0d0`) or an individual slice (`c0t0d0s7`)
- Recommended mode of operation: entire disk
  - No special formatting required
  - EFI label used to contain a single, large slice
  - Simplest way to create ZFS storage pools

The most basic element of a storage pool is a piece of physical storage. Physical storage can be any block device of at least 128 MB in size. Typically, this device is a hard drive that is visible to the system in the `/dev/dsk` directory.

A storage device can be a whole disk (`c1t0d0`) or an individual slice (`c0t0d0s7`). The recommended mode of operation is to use an entire disk, in which case the disk does not need to be specially formatted. ZFS formats the disk by using an Extensible Firmware Interface (EFI) label to contain a single, large slice.

Using whole physical disks is the simplest way to create ZFS storage pools. ZFS configurations become progressively more complex (from management, reliability, and performance perspectives) when you build pools from disk slices, Logical Unit Numbers (LUNs) in hardware RAID arrays, or volumes presented by software-based volume managers.

In this course, you use whole disks only.

ZFS Storage Pool Components: Disks

- To use whole disks:
  - Use `/dev/dsk/cXtXdX` naming convention.
  - Specify using either full path (`/dev/dsk/c1t0d0`) or shorthand name consisting of device name within the `/dev/dsk` directory (`c1t0d0`).
- Examples of valid disk names:
  - `c1t0d0`
  - `/dev/dsk/c1t0d0`

Disk Device File Syntax

`c#`: This device is attached to controller `#`.

`t#`: This is the target controller number that is physically stored on the SCSI disk. On a SCSI controller, this is the SCSI target ID and is usually set via a switch on any external enclosure or by jumpers on the disk itself.

`d#`: This is the disk number that is assigned by the system to a local disk.

ZFS Storage Pool Components: Slices

- Label disk with SMI label.
- For the bootable ZFS root pool:
  - Disk must contain slices
  - SMI label is required

Examples

- On SPARC-based system with 72-GB disk
  - 68 GB of usable space in slice 0
- On x86-based system with 72-GB disk:
  - 68 GB of usable space in slice 0
  - Small amount of boot information contained in slice 8
    - No administration required
    - Cannot change

ORACLE

Disks can be labeled with a traditional Solaris VTOC (SMI) label when you create a storage pool with a disk slice.

For a bootable ZFS root pool, the disks in the pool must contain slices and must be labeled with an SMI label. The simplest configuration would be to put the entire disk capacity in slice `0` and use that slice for the root pool.

For example, on a SPARC-based system with a 72-GB disk, you would need to have 68 GB of usable space located in slice `0`.

Similarly, on a x86-based system with a 72-GB disk, you would also need to allow 68 GB of usable space located in slice `0`. A small amount of boot information is contained in slice `8`. Slice `8` requires no administration and cannot be changed.

ZFS Storage Pool Components: Files

- Not intended for production use
- Recommended for:
  - Testing
  - Simple experimentation
- Complete file path specification required
- File size: At least 128 MB

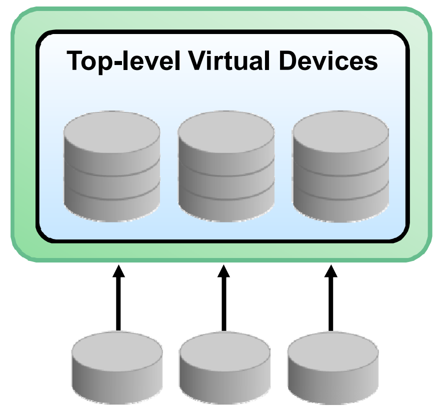
ZFS also allows you to use UFS files as virtual devices in your storage pool. This feature is not intended for production use. It is aimed primarily for testing and enabling simple experimentation. The reason is that any use of files relies on the underlying file system for consistency. If you create a ZFS pool backed by files on a UFS file system, you are implicitly relying on UFS to guarantee correctness and synchronous semantics.

However, files can be quite useful when you are first trying out ZFS or experimenting with more complicated layouts when not enough physical devices are present.

All files must be specified as complete paths and must be at least 128 MB in size.

ZFS Storage Pool Components: Virtual Devices

- A virtual device is a logical device in a pool:
  - Disks
  - Disks slices
  - Files
- Virtual devices at top of configuration are referred to as "top-level virtual devices" or "top-level vdevs."
- Possible configurations:
  - Stand-alone (non-redundant)
  - Mirrored
  - RAID-Z



Each ZFS storage pool is comprised of one or more virtual devices. A virtual device is a logical device in a pool that can be disks, disk slices, or files. A pool can have any number of virtual devices at the top of the configuration, known as top-level virtual devices or top-level vdevs.

You can configure these virtual devices to stand alone within a pool (referred to as an unreplicated or non-redundant configuration) or combine them into a mirror or RAID-Z virtual device to provide data redundancy.

Disks, disk slices, or files that are used in pools outside of mirrors and RAID-Z virtual devices, function as top-level virtual devices themselves.

Virtual Devices and Dynamic Striping

- Data is dynamically striped across all top-level virtual devices.
- Data placement is done at write time.
- When a new virtual device is added, data is gradually allocated to the new device.

Note: Although ZFS supports combining different types of virtual devices within the same pool, the recommended practice is to use top-level virtual devices of the same type with the same redundancy level in each device.

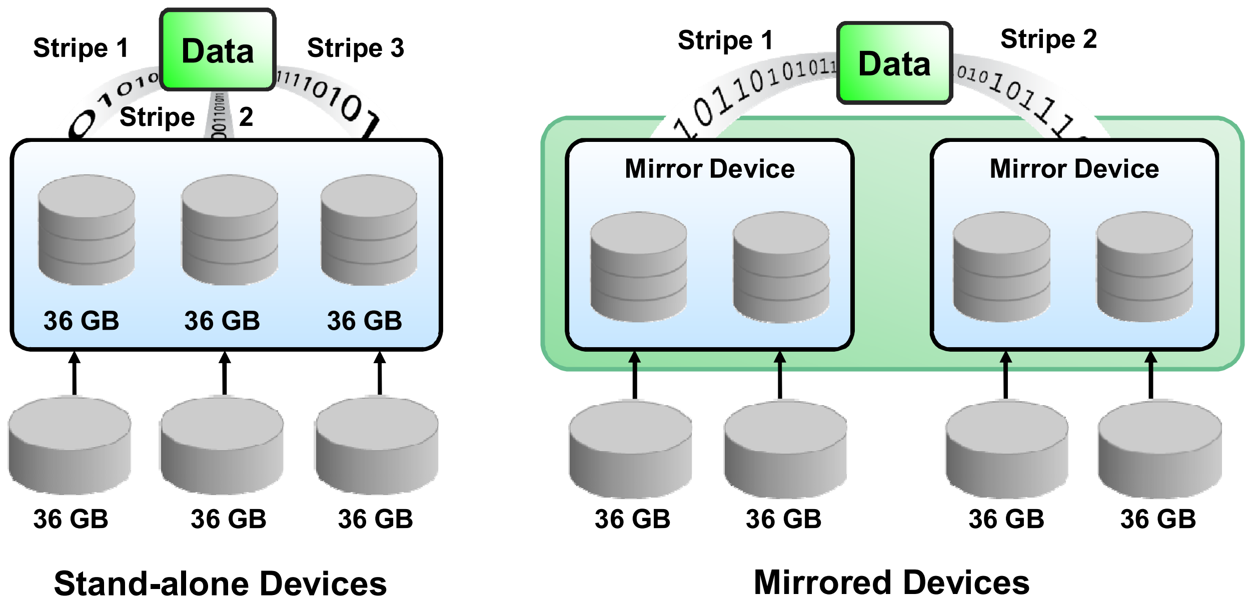
ZFS dynamically stripes data across all top-level virtual devices. The decision about where to place data is done at write time, so no fixed-width stripes are created at allocation time.

When new virtual devices are added to a pool, ZFS gradually allocates data to the new device to maintain performance and space allocation policies.

Note: Although ZFS supports combining different types of virtual devices within the same pool, this practice is not recommended. This is because your fault tolerance is only as good as your worst virtual device. The recommended practice is to use top-level virtual devices of the same type with the same redundancy level in each device.

Virtual Devices and Dynamic Striping

ZFS dynamically stripes data across all of the top-level virtual devices.



This slide shows two examples of how ZFS dynamically stripes data across all of the top-level virtual devices.

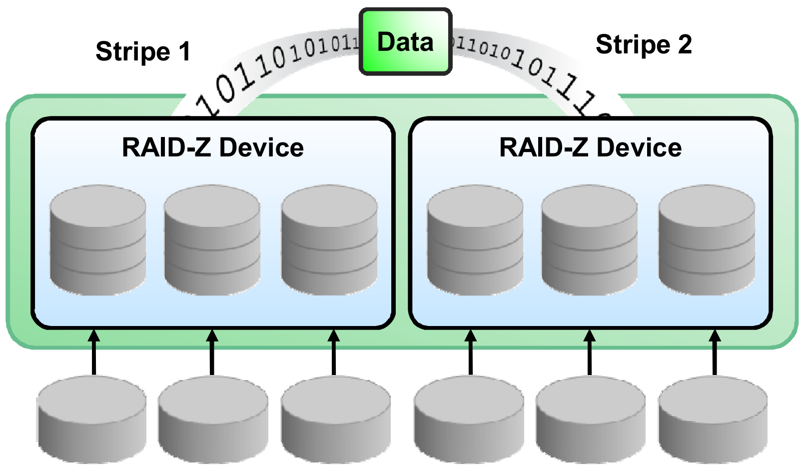
The diagram on the left shows how data is striped across three stand-alone disks, with each top-level virtual device mapped to a 36-GB disk. A ZFS pool that uses disks as top-level virtual devices, as shown in this diagram, provides no data redundancy.

The diagram on the right shows how data is striped across mirrored top-level virtual devices. In this configuration, there are two top-level mirrored virtual devices, each containing two disks of 36 GB. This configuration does provide data redundancy. A disk can be lost in either mirrored set and still not suffer any loss of data.

Virtual Devices and Dynamic Striping

Data is:

- Dynamically striped across all virtual devices in a RAID-Z pool
- Redundant within each virtual device in the RAID-Z pool



With a RAID-Z storage pool, the data is dynamically striped across all the virtual devices in the pool and redundant among the vdevs in the pool. For example, in the diagram shown here, data is being striped across two virtual devices, with three devices in one vdev and three in the other vdev. Data redundancy exists within each virtual device in the RAID-Z pool.

ZFS Storage Pool Data Redundancy

Data redundancy features:

- Mirrored storage pool configuration
  - Requires at least two disks
  - More than one mirror can be created in each pool.
- RAID-Z storage pool configuration
  - Single-parity RAID-Z (`raidz` or `raidz1`): Similar to RAID-5
  - Double-parity RAID-Z (`raidz2`): Similar to RAID-6
  - Triple-parity RAID-Z (`raidz3`): Similar to `raidz2` with an additional parity protection level
- Self-healing data
  - Supported in a mirrored or RAID-Z configuration
  - Automatically detects and repairs bad data blocks

ORACLE

Now that you are familiar with the ZFS storage pool components, you can look at ZFS's level of data redundancy features.

ZFS provides data redundancy through its mirrored and RAID-Z configuration options.

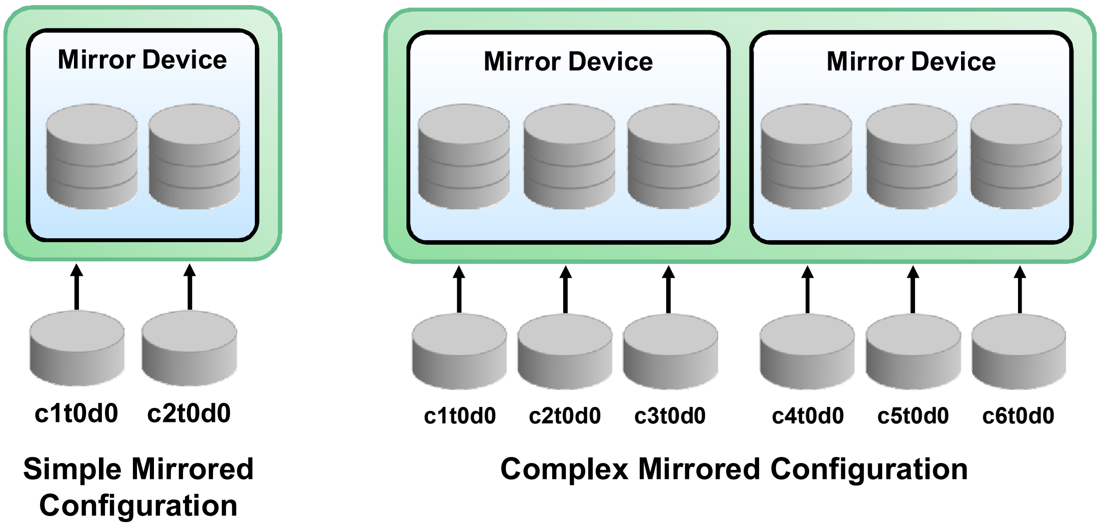
A mirrored storage pool configuration requires at least two disks, preferably on separate controllers. Many disks can be used in a mirrored configuration. In addition, you can create more than one mirror in each pool.

A RAID-Z storage pool can be a configuration with single-, double-, or triple-parity fault tolerance, which means that one, two, or three device failures can be sustained respectively without any data loss.

Single-parity RAID-Z (`raidz` or `raidz1`) is similar to RAID-5. Double-parity RAID-Z (`raidz2`) is similar to RAID-6. Triple-parity RAID-Z (`raidz3`) is similar to `raidz2` with an additional parity protection level.

ZFS provides for self-healing data in both mirrored and RAID-Z configurations. When a bad data block is detected, not only does ZFS fetch the correct data from another replicated copy, but it also repairs the bad data by replacing it with the good copy.

Mirrored Storage Pool Configuration: Examples



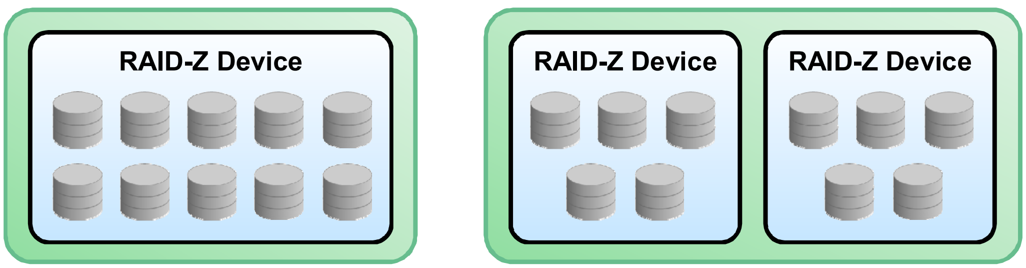
This slide shows examples of two mirrored storage pool configurations.

The diagram on the left shows an example of a simple mirrored configuration. The storage pool contains one mirror with two disks. In this example, you could lose only one disk before you start to lose data.

An example of a more complex mirrored configuration is shown in the diagram on the right. Here, the storage pool contains two mirrors with three disks each. With the more complex mirrored configuration example, you could lose up to two disks in each mirror and not lose any data.

RAID-Z Storage Pool Configuration: Examples

- Minimum disk usage recommendations by RAID-Z level
  - `raidz` or `raidz1`: Use at least three disks (2 + 1)
  - `raidz2`: Use at least five disks (3 + 2)
  - `raidz3`: Use at least eight disks (5 + 3)
- For better performance, configure storage pools with single-digit groupings of disks.



Not Recommended:
Large grouping of disks

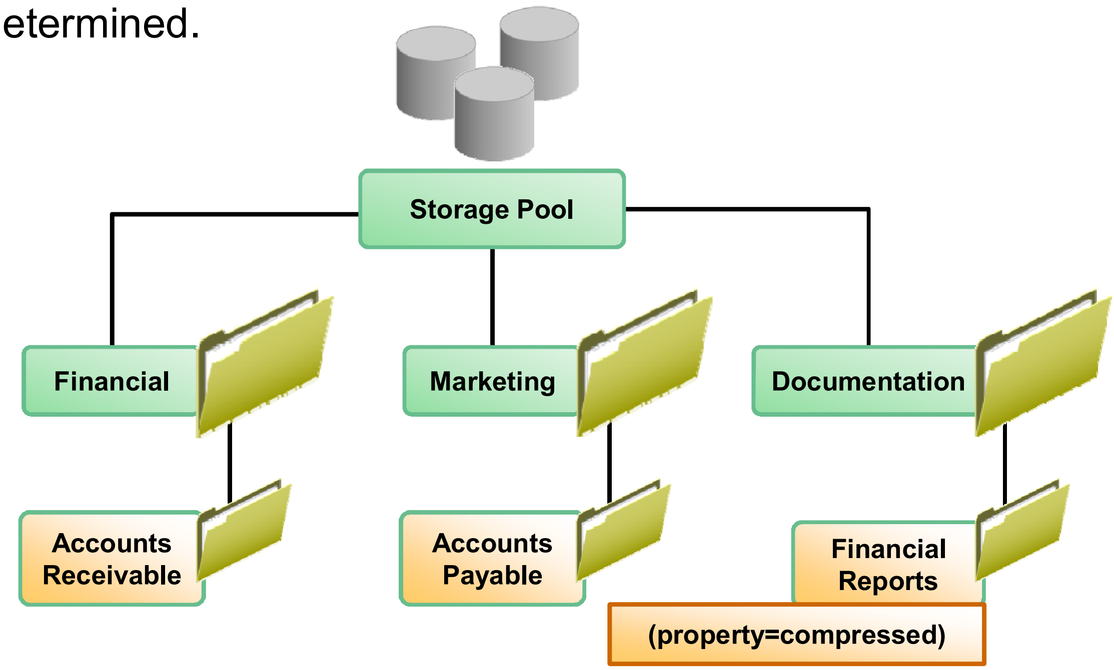
Recommended:
Single-digit grouping of disks

For a `raidz1` configuration, it is recommended that you use at least three disks (2 + 1). For a `raidz2` configuration, use at least five disks (3 + 2). For a `raidz3` configuration, use at least eight disks (5 + 3).

RAID-Z configurations with single-digit groupings of disks should perform better. Therefore, it is recommended that if you are creating a RAID-Z configuration with many disks, try dividing the disks into smaller groupings. For example, if you have a RAID-Z configuration with 14 disks, it would be better to split these 14 disks into two 7-disk groupings.

Determining ZFS File System Configuration Requirements

As part of planning, the file system configuration requirements are determined.



Determining how the file systems in a pool should be configured is another important part of data storage management planning. First, the file system granularity should be selected. A good model to use is a file system per-user or per-business application. This model allows properties, snapshots, and backups to be controlled on a per-user or per-business application basis.

Next, a decision should be made about how the file systems should be grouped. ZFS allows file systems to be organized into hierarchies so that similar file systems can be grouped. The root of the hierarchy is always the pool name. This model provides a central point of administration for controlling file system properties. Similar file systems should be created under a common name.

Finally, a determination should be made about which file system properties to set. Most file system characteristics are controlled by using simple properties. These properties control a variety of behaviors, including where the file systems are mounted, how they are shared, if they use compression, and if any quotas are in effect.

The diagram in the slide shows one storage pool with multiple file systems representing multiple business applications: a financial application, a marketing application, and a documentation application. Each application has been subdivided into sub-file systems, such as Accounts Receivables, Accounts Payables, and Financial Reports. To save storage space, the financial reports have been stored in a compressed format, using the ZFS `compression` property.

Note: In this course, you are introduced to the file system properties but you don't learn how to set them. Setting file system properties is covered in the *Oracle Solaris 11 Advanced System Administration* course.

Identifying Data Backup and Restore Requirements

As part of planning, backup and restore requirements are identified.

- Naming conventions for snapshots and clones
- Frequency with which snapshots and clones are taken
- Maintenance plan for snapshots and clones

Additional important considerations in data storage management planning include how the data will be backed up, how often it will be backed up, and how these backups can be used to recover data in case of data loss. ZFS provides these capabilities through its snapshot feature.

Knowing how snapshots and clones work and how they consume space as they change is important when dealing with data storage space concerns.

The data storage management plan should specify naming conventions for snapshots and clones, how often they are taken, and how they are maintained.

In this course, you learn how to create initial backups of your file systems by using snapshots. You also learn how to create clones to be used for creating new file systems from existing file systems.

Note: How to use snapshots to recover and restore data is covered in the *Oracle Solaris 11 Advanced System Administration* course.

Implementing the Data Storage Management Plan

Your assignment is to explore and test:
- ZFS storage pool functionality
- ZFS file system functionality
- ZFS snapshot and clone functionality

Now that you have been introduced to ZFS and have a better understanding of how ZFS supports the data storage needs of your company, it is time for you to receive your assignment. Your assignment is to explore and test the functionality of ZFS storage pools, file systems, and snapshots and clones.

Your tasks include:
- Creating and destroying different types of storage pools and displaying storage pool properties and status
- Creating, destroying, and mounting ZFS file systems as well as displaying file system properties
- Creating and destroying ZFS snapshots and clones

In the following sections, you learn the commands you need to perform these tasks.

Quiz

When working with RAIDZ pools, if you want better performance, you should configure the pools with single-digit groupings of disks.

a. True

b. False

Answer: a

Quiz

With a RAID-Z storage pool, the data is dynamically striped across all the virtual devices in the pool and redundant among the devices within a single vdev.

a. True
b. False

Answer: a

Quiz

Which RAID-Z parity options does ZFS support?

a. `raidz` / `raidz1` only

b. `raidz` / `raidz1` and `raidz2` only

c. `raidz` / `raidz1`, `raidz2`, and `raidz3`

d. `raidz` / `raidz1`, `raidz2`, `raidz3`, and `raidz4`

Answer: c

Lesson Agenda

- Planning for Data Storage Management
- Administering ZFS Storage Pools
- Administering ZFS File Systems
- Administering ZFS Snapshots and Clones

Now that the data storage management planning activities have been completed and you know what your assignment is, you learn how to administer the ZFS storage pools.

Creating ZFS Storage Pools

- Command: `zpool create`
  - Takes a pool name
  - Takes any number of virtual devices
- Types of pools:
  - Basic
  - Mirrored
  - RAID-Z

In this section, you will see that creating a ZFS storage pool is fast and easy. To create a storage pool, use the `zpool create` command, which takes a pool name and any number of virtual devices as arguments.

You can use the `zpool create` command to create several different types of storage pools: stand-alone (also referred to a non-redundant), mirrored, or RAID-Z.

You now look at how to create each pool type.

Creating a Basic Storage Pool

To create a basic ZFS pool, enter `zpool create` followed by the pool name and disks to include in the pool.

```
# zpool create hrpool c1t0d0 c1t1d0
```

Both disks are:

- Found in `/dev/dsk`
- Labeled by ZFS to contain a single, large slice
- Dynamically striped across with data

To create a basic storage pool, enter the `zpool create` command followed by the new pool name and the names of the disks you want in the pool. In the example command shown here, we've created a pool called `hrpool` using the `zpool create` command and the pool consists of two disks: `c1t0d0` and `c1t1d0`. These whole disks are found in the `/dev/dsk` directory and are labeled appropriately by ZFS to contain a single large slice. Data is dynamically striped across both disks.

Determining Local Storage Disk Availability

To display disk availability, run `format`.

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c7t0d0 <ATA-VBOX HARDDISK  -1.0  cyl 2085 alt 2 hd 255 sec 63>
          /pci@0,0/pci8086,2829@d/disk@0,0
       1. c7t2d0 <ATA-VBOX HARDDISK  -1.0  cyl 2085 alt 2 hd 255 sec 63>
          /pci@0,0/pci8086,2829@d/disk@2,0
       2. c7t3d0 <ATA-VBOX HARDDISK  -1.0  cyl 2085 alt 2 hd 255 sec 63>
          /pci@0,0/pci8086,2829@d/disk@3,0
       3. c7t4d0 <ATA-VBOX HARDDISK  -1.0  cyl 2085 alt 2 hd 255 sec 63>
          /pci@0,0/pci8086,2829@d/disk@4,0
       4. c7t5d0 <ATA-VBOX HARDDISK  -1.0  cyl 2085 alt 2 hd 255 sec 63>
        /pci@0,0/pci8086,2829@d/disk@5,0
Specify disk (enter its number):
```


You can determine what disks are available for use in a pool by using the `format` command. Recall using this command when you checked the disk configuration information during your operating system installation verification task. At that time, there was only one configured disk, `c7t0d0`, and that was being used by the root file system `rpool`.

The disks that you should use in each pool should be provided to you. More senior system or storage administrators are responsible for formatting the disks (if necessary) and making them available on the system.

If, by chance, you choose a disk that is unavailable and attempt to use it in the creation of a storage pool, the system lets you know.

For information about disk labels, disk formatting, and disk partitioning, see the “Managing Disks” section of *Oracle Solaris Administration: Devices and File Systems*.

Creating a Mirrored Storage Pool

To create a mirrored storage pool, enter `zpool create` followed by the pool name, the `mirror` keyword, and the storage devices that will comprise the mirror.

```
# zpool create hrpool mirror c1t0d0 c2t0d0 mirror c3t0d0 c4t0d0
```

Data is:

- Dynamically striped across both mirrors
- Redundant between each disk within a mirror

A mirrored storage pool is created using the `mirror` keyword, followed by any number of storage devices that will comprise the mirror. Multiple mirrors can be specified by repeating the `mirror` keyword on the command line.

The command shown in the example creates a pool called `hrpool` with two 2-way mirrors. The first mirror contains devices `c1t0d0` and `c2t0d0`, and the second mirror contains the devices `c3t0d0` and `c4t0d0`.

Data is dynamically striped across both mirrors, with data being redundant between each disk within a mirror.

Creating a RAID-Z Storage Pool

To create a ZFS RAID-Z storage pool, enter `zpool create` followed by the pool name, the `raidz` keyword, and the storage devices that will be part of each RAID-Z pool.

```
# zpool create hrpool raidz c1t0d0 c2t0d0 c3t0d0 c4t0d0
/dev/dsk/c5t0d0
```

```
# zpool create datapool raidz2 c1t0d0 c2t0d0 c3t0d0
c4t0d0 c5t0d0 c6t0d0 raidz2 c8t0d0 c9t0d0 c10t0d0
c11t0d0 c12t0d0 c13t0d0
```


Creating a RAID-Z storage pool is identical to creating a mirrored storage pool, except that the `raidz`, `raidz2`, or `raidz3` keyword is used instead of `mirror`.

In the first code example in the slide, a single-parity RAID-Z storage pool called `hrpool` has been created using the `zpool create` command. The pool consists of five disks: `c1t0d0`, `c2t0d0`, `c3t0d0`, `c4t0d0`, and `c5t0d0`. The `/dev/dsk/` path has been included for the `c5t0d0` disk to illustrate that disks can be specified using their full paths. The `/dev/dsk/c5t0d0` device is identical to the `c5t0d0` device.

You could also use disk slices in this configuration, but you would need to preformat the disks to have appropriately sized slices.

In the second code example, a double-parity RAID-Z storage pool has been created called `datapool`. The first `raidz2` virtual device contains six disks: `c1t0d0`, `c2t0d0`, `c3t0d0`, `c4t0d0`, `c5t0d0`, and `c6t0d0`. The second `raidz2` virtual device also contains six disks: `c8t0d0`, `c9t0d0`, `c10t0d0`, `c11t0d0`, `c12t0d0`, and `c13t0d0`.

Default Mount Point for Storage Pools

- Default mount point is `/pool-name`.
- Directory is automatically created if it does not exist.
- If directory exists, it must be empty.
- To change default mount point, use `–m` with `zpool create`.

```
# zpool create -m /export/zfs home c1t0d0
```


When a pool is created, the default mount point for the root dataset is `/pool-name`. This directory either must not exist or, if it does exist, must be empty. If the directory does not exist, ZFS automatically creates it. If the directory is empty, ZFS mounts the root dataset on top of the existing directory.

Note: If the directory is not empty, you will get an error message stating that the mount point exists and that it is not empty. The error message will direct you to use the `-m` option to provide a different default.

To create a pool with a different default mount point, use the `-m` option with the `zpool create` command.

For example, in the code sample shown here, the default mount point of the pool named `home` located on disk `c1t0d0` is being changed to a mount point of `/export/zfs`.

Now that you have a better idea of how different types of storage pools can be created using the `zpool create` command, spend just a moment examining how these pools can be destroyed.

Destroying a ZFS Storage Pool

To destroy a pool, enter `zpool destroy` followed by the pool name.

```
# zpool destroy testpool
```

Caution: Be very careful when you destroy a pool. Make sure you are destroying the right pool and that you always have copies of your data. If you accidentally destroy the wrong pool, you can attempt to recover the pool.

Destroying a pool is even easier than creating a pool. To destroy a pool, use the `zpool destroy` command followed by the pool name. This command destroys the pool even if it contains mounted datasets. In the example shown here, the pool named `testpool` is being destroyed.

Note: For more information about recovering a pool, refer to *Oracle Solaris Administration: ZFS File Systems.*

ZFS Storage Pool Properties

Pool properties :

- Determine the behavior of a pool feature, such as whether:
  - A pool is bootable
  - A property is enabled
- Identify read-only attributes, such as:
  - Current pool size
  - Unique pool identifier (GUID)

How a storage pool behaves is determined by its pool properties. Properties determine the behavior of a pool feature, such as whether the pool is bootable or whether a particular property is enabled.

For example, the `autoreplace` property controls automatic device replacement. If it is set to `off`, device replacement must be initiated by using the `zpool replace` command. If it is set to `on`, any new device that is found in the same physical location as a device that previously belonged to the pool is automatically formatted and replaced. The default behavior is `off`.

A pool property can also identify a read-only attribute, such as the current pool size or the unique identifier for the pool, the GUID.

Displaying Pool Properties

Use `zpool get all` followed by the pool name to display all the property information for a pool.

```
# zpool get all assetpool
NAME       PROPERTY         VALUE                          SOURCE
assetpool  size             33.8G                          -
assetpool  capacity         17%                            -
assetpool  altroot          -                              default
assetpool  health           ONLINE                         -
assetpool  guid             17361998391267837263           -
assetpool  version          33                             default
assetpool  bootfs           assetpool/ROOT/solaris             local
assetpool  delegation       on                             default
assetpool  autoreplace      off                            default
assetpool  cachefile        -                              default
assetpool  failmode         wait                           local
assetpool  listsnapshots    off                            default
assetpool  autoexpand       off                            default
assetpool  dedupditto       0                              default
assetpool  depupratio       1.00x                          -
assetpool  free             27.8G                          -
assetpool  allocated        5.91G                          -
assetpool  readonly         off                            -
```


You can display all the property information for a pool by using the `zpool get all` command followed by the pool name.

In the example shown here, the property information is being displayed for a pool called `assetpool`. As you can see, the property types are listed in the `PROPERTY` column, the values associated with each property are listed in the `VALUE` column, and the property source is listed in the `SOURCE` column.

Take a few minutes to familiarize yourself with the pool properties that you will be seeing most often in this class. For the most complete and up-to-date listing and description of the ZFS pool properties, see *Oracle Solaris Administration: ZFS File Systems*.

Displaying Pool Properties

```
# zpool get all assetpool
NAME       PROPERTY       VALUE                         SOURCE
assetpool  size           33.8G                         -
assetpool  capacity       17%                           -
assetpool  altroot        -                             default
assetpool  health         ONLINE                        -
assetpool  guid           17361998391267837263          -
assetpool  version        33                            default
assetpool  bootfs         assetpool/ROOT/solaris            local
assetpool  delegation     on                            default
assetpool  autoreplace    off                           default
assetpool  cachefile      -                             default
assetpool  failmode       wait                          local
assetpool  listsnapshots  off                           default
assetpool  autoexpand     off                           default
assetpool  dedupditto     0                             default
assetpool  depupratio     1.00x                         -
assetpool  free           27.8G                         -
assetpool  allocated      5.91G                         -
assetpool  readonly       off                           -
```


The first property you see in the example is `size`. This is a read-only property that identifies the total size of the storage pool. In this example, the pool is 33.8 GB in size.

Next you see `capacity`. This read-only value identifies the percentage of pool space used. This property can also be referred to by its shortened column name, `cap`.

Next you see the `health` property. This read-only property identifies the current health of the pool, as `ONLINE`, `DEGRADED`, `FAULTED`, `OFFLINE`, `REMOVED`, or `UNAVAIL`. In this example, `assetpool` has a health status of `ONLINE`.

You have already learned a bit about the next property, which is `autoreplace`. As you may recall, this property controls automatic device replacement. As previously mentioned, the default behavior is `off`, which is the setting seen in this example. This property can also be referred to by its shortened column name, `replace`.

Displaying Pool Properties

```
# zpool get all assetpool
NAME       PROPERTY         VALUE                         SOURCE
assetpool  size             33.8G                         -
assetpool  capacity         17%                           -
assetpool  altroot          -                             default
assetpool  health           ONLINE                        -
assetpool  guid             17361998391267837263          -
assetpool  version          33                            default
assetpool  bootfs           assetpool/ROOT/solaris            local
assetpool  delegation       on                            default
assetpool  autoreplace      off                           default
assetpool  cachefile        -                             default
assetpool  failmode         wait                          local
assetpool  listsnapshots    off                           default
assetpool  autoexpand       off                           default
assetpool  dedupditto       0                             default
assetpool  depupratio       1.00x                         -
assetpool  free             27.8G                         -
assetpool  allocated        5.91G                         -
assetpool  readonly         off                           -
```


The last four properties focused on are `listsnapshots`, `autoexpand`, `free`, and `allocated`.

The `listsnapshots` property controls whether snapshot information that is associated with this pool is displayed with the `zfs list` command. The default value is `off`. You will look at snapshots in more detail a bit later in the course. In this example, the `listsnapshots` property is set to the default, so it is `off`.

Using the `autoexpand` property, you can enable or disable automatic pool expansion when a dynamic LUN expansion event is received. This property is disabled by default (as shown in this example), so you can decide whether you want the LUN expanded or not.

The `free` property is a read-only value that identifies the amount of storage that is available within the pool. In this example, there is 27.8 GB of space available in the pool named `assetpool`.

The `allocated` property identifies the amount of storage space used within the pool. It is also a read-only property. In this example, the pool is currently using only 5.91 GB of space.

Querying ZFS Pool Status

You can request the following types of information about a pool:

- Basic usage information
- I/O statistics
- Health status
- Command history

You can request different types of information regarding pool status. The information you can obtain generally falls into three categories: basic pool usage information, pool I/O statistics, and pool health status. You can also display the command history for a pool.

In this section, you learn the commands for accessing all four types of information, beginning with basic pool usage.

Displaying Basic Pool Usage Information

To display usage information about all pools on the system, use `zpool list`.

```
# zpool list
NAME       SIZE   ALLOC   FREE   CAP  DEDUP  HEALTH   ALTROOT
hrpool    80.0G  22.3G  47.7G   28%  1.00x  ONLINE   -
datapool   1.2T  384G    816G   32%  1.00x  ONLINE   -
```

Note: To gather statistics for a specific pool, specify the pool name, as in the following example: `zpool list hrpool`.

ORACLE

To display basic pool usage information, use the `zpool list` command. You can use this command without any arguments to display usage information about all pools on the system.

The example here shows there are two pools on the system: `hrpool` and `datapool`. The usage information provided for both pools includes the size of the pool (80 GB and 1.2 TB, respectively), the amount of space allocated (22.3 GB and 384 GB, respectively), the amount of space available or free (47.7 GB and 816 GB, respectively), the percentage of pool space used (that is, the capacity) (28% for `hrpool` and 32% for `datapool`), the deduplication ratio specified for the pool, the current health status of the pool (both pools are online), and the alternate root directory (neither pool has an alternate root directory).

Note: The deduplication property controls the ability to remove duplicate data in a ZFS file system.

Displaying Specific Pool Statistics

To display specific statistics, use `zpool list` with the `-o` option.

```
# zpool list -o name,size
NAME       SIZE
hrpool     80.0G
datapool   1.2T
```


You can request specific statistics by using the `-o` option with the `zpool list` command. For example, if you want to see only the name and size of each pool, enter the `zpool list` command followed by the `-o` option and then the words `name` and `size` separated by a comma (as shown in this example).

Displaying Specific Pool Statistics

Use the `-H` option to suppress column headings and to separate fields by tabs rather than by spaces.

```
# zpool list -H -o name,size
hrpool    80.0G
datapool  1.2T
```


Because the default output for the `zpool list` command is designed for readability and is not easy to use as part of a shell script, you can use the `-H` option to suppress column headings and to separate fields by tabs rather than by spaces.

In the example shown here, the `-H` option has been used to suppress the `NAME` and `SIZE` headings for the usage information about the `hrpool` and `datapool` pools.

Now that you have looked at how to use the `zpool list` command to display basic pool usage information, take a look at the command used to view pool I/O statistics.

Viewing Pool I/O Statistics

Use `zpool iostat` with no options to display the accumulated statistics since boot for all pools on the system.

```
# zpool iostat
                 capacity        operations        bandwidth
pool            alloc  free      read   write      read   write
----------      -----  -----     -----  -----      -----  -----
hrpool          100G   20.0G     1.2M   102K       1.2M   3.45K
datapool        12.3G  67.7G     132K   15.2K      32.1K  1.20K
----------      -----  -----     -----  -----      -----  -----
```

Note: You can request a more accurate view of current bandwidth usage by specifying an interval.

To request I/O statistics, use the `zpool iostat` command. Similar to the Solaris `iostat` command, this command can display a static snapshot of all I/O activity so far, as well as updated statistics for every specified interval.

With no options, the `zpool iostat` command displays the accumulated statistics since boot for all pools on the system.

In this example, the `zpool iostat` command has been run for the two pools in the system. The output from the command provides you with the capacity (allocated and free) for each pool as well as the statistics for the read/write operations of each pool, and the amount of bandwidth used for the read/write operations, expressed as units per second.

Note: Because these statistics are cumulative since boot, bandwidth might appear low if the pool is relatively idle. You can request a more accurate view of current bandwidth usage by specifying an interval. For more information about using intervals with the `zpool iostat` command, see “Listing Pool-Wide Statistics” in *Oracle Solaris Administration: ZFS File Systems.*

Viewing Pool I/O Statistics

Use `zpool iostat -v` to request the complete virtual device layout as well as all I/O statistics.

```
# zpool iostat -v hrpool
                  capacity        operations        bandwidth
hrpool          alloc  free      read  write      read  write
----------      -----  -----     -----  -----     -----  -----
mirror          20.4G  59.6G         0     22         0  6.00K
  c1t0d0            -      -         1    295     11.2K   148K
  c1t1d0            -      -         1    299     11.2K   148K
----------      -----  -----     -----  -----     -----  -----
```


In addition to pool-wide I/O statistics, you can use the `zpool iostat` command to display statistics for virtual devices. This command can be used to identify abnormally slow devices, or simply to observe the distribution of I/O generated by ZFS. To request the complete virtual device layout as well as all I/O statistics, use the `zpool iostat -v` command.

In the example shown here, look at the I/O statistics for the virtual devices in the mirrored pool named `hrpool`: `c1t0d0` and `c1t1d0`. With this view you can see the read/write operation values and read/write bandwidth values for both devices. You can also see the pool totals for each statistic.

Viewing Pool I/O Statistics

When viewing I/O statistics on a virtual device, remember the following:

- Space usage is available for top-level virtual devices only.
- The numbers might not always add up as you expect.
  - Most noticeable immediately after pool creation
  - Should gradually equalize
  - Broken, unresponsive, or offlined devices can affect symmetry as well.

Note: You can also specify intervals when examining virtual device statistics.

When viewing I/O statistics on a virtual device basis, there are two things you need to remember.

First, space usage is available for top-level virtual devices only.

Second, the numbers might not add up exactly as you would expect. In particular, operations across RAID-Z and mirrored devices will not be exactly equal. This difference is particularly noticeable immediately after a pool is created, as a significant amount of I/O is done directly to the disks as part of pool creation that is not accounted for at the mirror level. Over time, these numbers should gradually equalize, although broken, unresponsive, or offlined devices can affect this symmetry as well.

Now that you have looked at how to use the `zpool iostat` command to view both pool-wide I/O statistics and the virtual devices within a pool, look at the command used to determine the health status of a pool.

Determining the Health Status of a Pool

- The health of a pool is:
  - Determined from the state of all its devices
  - Displayed by using the `zpool status` command
- Potential pool and device failures are:
  - Reported by `fmd`
  - Displayed on the system console
  - Logged in the `/var/adm/messages` file

ZFS provides an integrated method of examining pool and device health. The health of a pool is determined from the state of all its devices. This state information is displayed by using the `zpool status` command.

In addition, potential pool and device failures are reported by the fault management daemon (`fmd`) and are displayed on the system console and logged in the `/var/adm/messages` file.

Determining the Health Status of a Pool

Each device can fall into one of the following states:

- `ONLINE`: Device or virtual device is in normal working order.
- `DEGRADED`: Virtual device has experienced failure but is still able to function.
- `FAULTED`: Device or virtual device is completely inaccessible.
- `OFFLINE`: Device has been explicitly taken offline by the administrator.
- `REMOVED`: Device was physically removed while the system was running.
- `UNAVAIL`: Device or virtual device cannot be opened (that is, unavailable).

Determining the Health Status of a Pool

The health of a pool is determined from the health of all its top-level virtual devices:

State of Top-Level vdevs	State of Pool	Result
All vdevs online	`ONLINE`	No issues are present.
One or more vdevs degraded or unavailable	`DEGRADED`	Pool continues to run but level of redundancy or data throughput might be affected.
One or more vdevs faulted or offline	`FAULTED`	Pool is completely inaccessible. No data recovery is possible until devices are attached or repaired.

The health of a pool is determined from the health of all its top-level virtual devices.

If all virtual devices are `ONLINE`, then the pool is also `ONLINE`.

If any one of the virtual devices is `DEGRADED` or `UNAVAIL`, then the pool is also `DEGRADED`. A pool in the degraded state continues to run, but you might not achieve the same level of data redundancy or data throughput than if the pool were online.

If a top-level virtual device is `FAULTED` or `OFFLINE`, then the pool is also `FAULTED`. A pool in the faulted state is completely inaccessible. No data can be recovered until the necessary devices are attached or repaired.

Determining the Health Status of a Pool

Use `zpool status -x` to request a quick overview of pool health status.

```
# zpool status -x
all pools are healthy
```


The simplest way to request a quick overview of pool health status is to use the `zpool status` command followed by the `-x` option, as shown in the example in the slide. In this example, all the pools on the system are healthy.

Determining the Health Status of a Pool

To examine the health of a specific pool, use `zpool status` followed by the pool name.

```
# zpool status hrpool
  pool: hrpool
 state: ONLINE
 scrub: non requested
config:
        NAME            STATE     READ   WRITE   CKSUM
        hrpool          ONLINE       0       0       0
          mirror-0      ONLINE       0       0       0
            c1t0d0      ONLINE       0       0       0
            c1t1d0      ONLINE       0       0       0

errors: No known data errors
```


You can examine the health of a specific pool by specifying a pool name following the `zpool status` command. For example, if you want to know the health of the pool named `hrpool`, enter the `zpool status` command followed by the name `hrpool`, as shown in this example.

Determining the Health Status of a Pool

Use `zpool status -v` followed by the pool name to request a more detailed summary of a pool's health status.

```
# zpool status -v hrpool
  pool: hrpool
 state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas
    exist for the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: none requested
config:
        NAME          STATE       READ    WRITE   CKSUM
        hrpool        DEGRADED       0        0       0
          mirror-0    DEGRADED       0        0       0
            c1t0d0    FAULTED        0        0       0  cannot open
            c1t1d0    ONLINE         0        0       0
errors: No known data errors
```


If the results of the `zpool status` command returned anything other than a healthy, online state, you should investigate what the problem is. You can request a more detailed summary of the health status by using the `-v` option with the `zpool status` command, as shown in the example.

Notice that in this example, the mirrored pool named `hrpool` is in a degraded state because device `c1t0d0` has faulted and cannot be opened.

Notice also that, according to the status message, although the pool is in a degraded state, it can continue to function because sufficient replicas exist. In other words, the second disk in the pool is still online; however, there is no longer any data redundancy in the pool with just one functioning device left.

Notice also the action message, which recommends that the missing device be attached and brought back online using the `zpool online` command, and the URL after `see`, which provides additional information about how to resolve the issue.

Using the detailed configuration information provided here, you should be able to determine which device is damaged and how to repair the pool.

Displaying Pool Command History

Use `zpool history` to display `zpool` commands that modify pool state information.

```
# zpool history
History for 'hrpool':
2011-09-23.11:20:57 zpool create hrpool c1t0d0 c2t0d0
```

ORACLE

Now, take a brief look at how to display the command history for a ZFS storage pool. ZFS automatically logs successful `zpool` commands that modify pool state information. You can display this information by using the `zpool history` command (as shown in this example).

From the output in the example, you can see that on September 23, 2011, at 11:20:57, the pool named `hrpool` was created using two disks: `c1t0d0` and `c2t0d0`.

Quiz

What command is used to create a ZFS storage pool?

a. `zpool start new pool`

b. `zpool storagepool`

c. `zpool create`

d. `zpool make`

Answer: c

Quiz

After you have created a pool, you must manually create the mount point for the pool.

a. True

b. False

Answer: b

Quiz

What command is used to display all the property settings within a pool?

a. `zpool show all` <*poolname*>

b. `zpool get all` <*poolname*>

c. `zpool display all` <*poolname*>

d. `zpool set all` <*poolname*>

Answer: b

Quiz

What command is used to display basic pool usage information?

a. `zpool list`

b. `zpool iostat`

c. `zpool history`

d. `zpool status`

Answer: a

Quiz

If a pool is in a `DEGRADED` state, the data is completely inaccessible.

a. True

b. False

Answer: b

Practice 5-1 Overview: Administering ZFS Storage Pools

This practice covers the following topics:

- Creating different types of ZFS pools
- Querying the pool attributes

In the practices for this lesson, you are presented with four tasks designed to reinforce the concepts presented in the lecture portion of this lesson. You will have the chance to perform the following tasks:

- **Practice 5-1:** Administering ZFS storage pools
- **Practice 5-2:** Administering ZFS file systems
- **Practice 5-3:** Administering ZFS snapshots and clones
- **Practice 5-4:** Administering ZFS storage pools with disk slices

You will find Practice 5-1 in your *Activity Guide*. It should take you about 20 minutes to complete.

Lesson Agenda

- Planning for Data Storage Management
- Administering ZFS Storage Pools
- Administering ZFS File Systems
- Administering ZFS Snapshots and Clones

Administering ZFS File Systems

- ZFS file system: overview
- Creating, destroying, and renaming a ZFS file system
- Querying ZFS properties
- Displaying basic ZFS file system information
- Mounting and unmounting ZFS file systems

In this section, you learn how to create, destroy, and rename a ZFS file system. You are introduced to the ZFS properties, and you learn how to query them. Next, you learn how to display basic ZFS file system information. Finally, you learn how to mount and unmount ZFS file systems.

Creating a ZFS File System

To create a file system, enter `zfs create` followed by the file system path name.

```
# zfs create hrpool/home/reports
```

The file system name is specified as a path name:

`pool-name/[filesystem-name/]filesystem-name`

The pool name and initial file system names identify the location in the hierarchy where a new file system will be created.

The last name identifies the file system to be created.

Note: You can create missing intermediate file system names automatically by using the `zfs create -p command`.

You use the `zfs create` command to create ZFS file systems. The `create` subcommand takes a single argument: the name of the file system to create. In the example shown here, a file system named `reports` is being created in the `hrpool/home` file system.

The file system name is specified as a path name starting from the name of the pool:

`pool-name/[filesystem-name/]filesystem-name`

The pool name and initial file system names in the path identify the location in the hierarchy where the new file system will be created. The last name in the path identifies the name of the file system to be created.

Note: You can create any non-existent, intermediate file system names automatically by using the `-p` option with the `zfs create` command.

Creating a ZFS File System

File system mounted:

- Automatically if created successfully
- As `/dataset`
- Using path provided in the `create` subcommand

If you have created the file system successfully, ZFS automatically mounts it. By default, the file system is mounted as `/dataset`, using the path you provided for the file system name in the `create` subcommand.

Destroying a ZFS File System

To destroy a file system, enter `zfs destroy` followed by the file system path name.

```
# zfs destroy hrpool/home/oldreports
```

Caution: No confirmation prompt appears with the `destroy` subcommand.

To destroy a ZFS file system, use the `zfs destroy` command followed by the file system path name. In the example shown here, the file system named `oldreports` is being destroyed in `hrpool/home`.

Caution: Because no confirmation prompt appears with the `destroy` subcommand, use it with extreme caution. Unlike the `zpool destroy` command, the `zfs destroy` command cannot be reversed and the file system cannot be recovered.

Destroying a ZFS File System

If the `zfs destroy` command fails, use one of the options shown here.

Condition	Option	Results
File system is busy.	`-f`	This option can unmount, unshare, and destroy active file systems, causing unexpected application behavior.
File system has children.	`-r`	Recursively destroys a file system and all its descendents. This option also destroys snapshots.
File system has indirect dependents.	`-R`	Forces the destruction of all dependents, including cloned file systems outside the target hierarchy.

Caution: No confirmation prompts appear with the `-f`, `-r`, and `–R` options.

If the `zfs destroy` command fails, there are several options you can use to force the destruction of the file system. If the file system to be destroyed is busy and thus cannot be unmounted, you can use the `-f` option. Use this option with caution because it can unmount, unshare, and destroy active file systems, causing unexpected application behavior.

The `zfs destroy` command also fails if a file system has children. To recursively destroy a file system and all its descendents, use the `-r` option. Note that a recursive destroy also destroys snapshots, so you should use this option with caution.

If the file system to be destroyed has indirect dependents, even the recursive destroy command described above fails. To force the destruction of *all* dependents, including cloned file systems outside the target hierarchy, the `-R` option must be used.

Now that you are more familiar with how to create and destroy a ZFS file system, take a look at how you can rename a file system.

Renaming a ZFS File System

To rename a file system, enter `zfs rename` followed by the file system path name.

```
# zfs rename hrpool/home/reviews hrpool/home/reviews_2011
```

You can use the rename subcommand to:

- Change the name of a file system
- Relocate the file system to a new location within the ZFS hierarchy
- Change the name of a file system and relocate it within the ZFS hierarchy

ORACLE

You can rename a file system by using the `zfs rename` command. In the example shown here, you are taking the `reviews` file system that resides in `hrpool/home` and doing a simple rename to `reviews_2011`.

Renaming a ZFS File System

Example of file system relocation:

```
# zfs rename hrpool/home/jobdesc hrpool/ws/jobdesc
```

- New location must:
  - Be within the same pool
  - Have enough space to hold the new file system

The example here shows how to use `zfs rename` to relocate a file system. As you can see, the `jobdesc` file system is being relocated from `hrpool/home` to `hrpool/ws`.

When you relocate a file system through rename, the new location must be within the same pool and it must have enough space to hold this new file system. If the new location does not have enough space, possibly because it has reached its quota, the rename fails.

Renaming a ZFS File System

- The rename operation attempts an unmount/remount sequence for:
  - The file system
  - Any descendent file systems
- If unable to unmount an active file system:
  - Rename operation fails
  - Forced unmount is required

Querying ZFS Properties

- ZFS properties: overview
- Types of native ZFS properties
- Querying ZFS properties

In this section, you are introduced to the ZFS file system properties, the types of properties, and their uses. You also learn how they can be queried.

You now take a closer look at each of these topics, beginning with a brief overview of the ZFS file system properties.

ZFS Properties

- Properties allow you to control:
  - File systems
  - Volumes
  - Snapshots
  - Clones
- Two property types:
  - Native
    - Export internal statistics
    - Control ZFS file system behavior
  - User defined
    - No effect on ZFS file system behavior
    - Can be used to annotate datasets

Properties provide the main mechanism that you use to control the behavior of file systems, volumes, snapshots, and clones.

Native properties either export internal statistics or control ZFS file system behavior.

User properties have no effect on ZFS file system behavior, but you can use them to annotate datasets in a way that is meaningful in your environment.

This course focuses primarily on the native properties. If you want to find out more about user properties, see the "ZFS User Properties" section in *Oracle Solaris Administration: ZFS File Systems*

Types of Native ZFS Properties

- Read-only statistics
  - Can be retrieved but not set
  - Are not inherited
- Settable
  - Can be both retrieved and set
  - Most are inheritable (exceptions: quotas and reservations)

Note: An inheritable property is a property that, when set on a parent, is propagated to all of its descendents.

Quotas and reservations are covered in the *Oracle Solaris 11 Advanced System Administration* course.

Identifying Native ZFS Properties

Property Name	Type	Default Value	Description
`compression`	String	`off`	Enables or disables compression for a dataset
`mountpoint`	String	N/A	Controls the mount point used for this file system
`quota`	Number (or none)	`none`	Limits the amount of disk space that a dataset and its descendents can consume
`readonly`	Boolean	`off`	Controls whether a dataset can be modified. When it is set to `on`, no modifications can be made.
`sharenfs`	String	`off`	Controls whether a ZFS dataset is published as an NFS share

ORACLE

The table in the slide displays a sampling of the native ZFS properties, some of which you will encounter in this course. For a complete list and full descriptions of the native ZFS properties, see the "Introducing ZFS Properties" section of *Oracle Solaris Administration: ZFS File Systems*.

Listing Basic ZFS Information

To display basic dataset information, use `zfs list` with no options.

```
# zfs list
NAME                   USED   AVAIL  REFER     MOUNTPOINT
pool                   476K   16.5G    21K     /pool
pool/clone             18K    16.5G    18K     /pool/clone
pool/home              296K   16.5G    19K     /pool/home
pool/home/data         277K   16.5G   277K     /pool/home/data
pool/test               18K   16.5G    18K     /test
```


The `zfs list` command displays the names of all datasets on the system, including their `used`, `available`, `referenced`, and `mountpoint` properties (as illustrated in the example shown here).

Listing Basic ZFS Information

You can also use `zfs list` to display:

- Specific datasets by using the dataset name
- Dataset descendents recursively with `-r`

```
# zfs list -r pool/home/data
NAME                   USED   AVAIL  REFER    MOUNTPOINT
pool/home/data         277K   16.5G   277K    /pool/home/data
```


In the example shown in the slide, the basic information for `pool/home/data` is listed using the `-r` option.

Mounting and Unmounting ZFS File Systems

- Mounting ZFS file systems
- Unmounting ZFS file systems

In this section, you learn how to mount and unmount ZFS file systems.

Mounting ZFS File Systems

Use the `zfs mount` command to:

- View ZFS managed mounted file systems
- Change mount options
- Explicitly mount a file system

By default, all ZFS file systems are mounted by ZFS at boot by using the Service Management Facility (SMF) `svc:/system/filesystem/local` service. File systems are mounted under */path*, where *path* is the name of the file system.

However, you can use the `zfs mount` command to view the currently mounted ZFS managed file systems, change mount options, or explicitly mount a file system.

Mounting ZFS File Systems

To view all file systems currently mounted and managed by ZFS, use `zfs mount` with no arguments.

```
# zfs mount
hrpool /hrpool
hrpool/home /hrpool/home
hrpool/home/reports /hrpool/home/reports
```

To mount all ZFS managed file systems, use `zfs mount -a`.

```
# zfs mount -a
```

Note: A mounted file system uses a set of mount options based on the property values associated with the dataset.

The `zfs mount` command with no arguments shows all currently mounted file systems that are managed by ZFS. Note that the legacy managed mount points are not displayed.

You can use the `-a` option with the `zfs mount` command to mount all ZFS managed file systems. Legacy managed file systems are not mounted.

Note: When a file system is mounted, it uses a set of mount options based on the property values associated with the dataset. If any of these mount options are set explicitly by using the `-o` option with the `zfs mount` command, the associated property value is temporarily overridden. These property values are reported as temporary by the `zfs get` command and revert to their original settings when the file system is unmounted. If a property value is changed while the dataset is mounted, the change takes effect immediately, overriding any temporary setting.

Now, you take a look at how to unmount a ZFS file system.

Unmounting a ZFS File System

To unmount a ZFS file system, use `zfs unmount` followed by either the file system name or mount point.

```
# zfs unmount hrpool/home/qarpts
```

```
# zfs unmount /export/home/qarpts
```

Note: If the file system is active or busy, `zfs unmount` fails. You can use `-f` to force the unmount, but you should use this option with caution.

You can unmount file systems by using the `zfs unmount` subcommand. This command can take either the file system name or the mount point as arguments.

In the first example, the file system `hrpool/home/qarpts` is being unmounted by file system name.

In the second example, same file system is being unmounted by mount point name (`/export/home/qarpts`).

Note: The `unmount` command fails if the file system is active or busy. To forcibly unmount a file system, you can use the `-f` option. Be cautious when forcibly unmounting a file system if its contents are actively being used.

Quiz

What command is used to create a ZFS file system?

a. `zfs make`

b. `zfs create`

c. `zpool create`

d. `zpool make`

Answer: b

Quiz

What option, when used with the `zfs destroy` command, can destroy an active ZFS file system?

a. `-a`

b. `-f`

c. `-r`

d. `-R`

Answer: b

Quiz

When you relocate a file system through `rename`, the new location must be within the same pool.

a. True

b. False

Answer: a

Practice 5-2 Overview: Administering ZFS File Systems

This practice covers the following topics:

- Creating a ZFS file system
- Destroying a ZFS file system

This practice should take you about 15 minutes to complete.

Lesson Agenda

- Planning for Data Storage Management
- Administering ZFS Storage Pools
- Administering ZFS File Systems
- Administering ZFS Snapshots and Clones

Administering ZFS Snapshots and Clones

- Creating, destroying, and displaying ZFS snapshots
- Creating and destroying ZFS clones

In this section, you learn how to create, destroy, and display snapshots. You also learn how to create and destroy clones.

Creating a ZFS Snapshot

To create a snapshot, enter `zfs snapshot` followed by the snapshot name.

The snapshot name is specified as follows:

- *`filesystem@snapname`*
- *`volume@snapname`*

```
# zfs snapshot hrpool/home/reports@friday
```


`zfs snapshot` takes as its only argument the name of the snapshot to create.

In the example shown here, you are creating a snapshot of `hrpool/home/reports` that is named `hrpool/home/reports@friday`. This snapshot can now serve as a backup for the file system in case you need to restore the file system for any reason.

Creating a ZFS Snapshot

To create snapshots for all descendent file systems, use `zfs snapshot -r` and the snapshot name.

```
# zfs snapshot -r hrpool/home@now
# zfs list -t snapshot
NAME                          USED  AVAIL  REFER    MOUNTPOINT
hrpool/home@now                  0      -  29.5K    -
hrpool/home/reports@now          0      -  2.15M    -
hrpool/home/reviews@now          0      -  1.89M    -
hrpool/home/jobdesc@now          0      -  1.89M    -
hrpool/home/bonus@now            0      -  2.15M    -
```

Note: Snapshots have no modifiable properties, and dataset properties cannot be applied to a snapshot.

You can create snapshots for all descendent file systems by using the `zfs snapshot` command with the `-r` option followed by the snapshot name.

With the first command shown in the example, you are creating a snapshot for all the descendent file systems of `hrpool/home` and calling the snapshot `hrpool/home@now`.

With the second command, `zfs list -t snapshot`, you are displaying the ZFS snapshots. The `-t` option is used to display snapshot information. You will take a closer look at this option when learning about displaying and accessing snapshots.

Now, you look at how to destroy a ZFS snapshot.

Destroying a ZFS Snapshot

To destroy a snapshot, use `zfs destroy` followed by the snapshot name.

```
# zfs destroy hrpool/home/reports@now
```

Things to know when attempting to destroy a snapshot:

- The dataset cannot be destroyed if snapshots of it exist.
- Clones created from a snapshot must be destroyed before the snapshot can be destroyed.

In the example shown here, you are destroying the snapshot named `hrpool/home/reports@now`.

Displaying a ZFS Snapshot

To display snapshots, enter `zfs list -t snapshot`.

```
# zfs list -t snapshot
NAME                            USED  AVAIL  REFER  MOUNTPOINT
hrpool/home/reports@tuesday      18K      -    21K  -
hrpool/home/reports@wednesday    19K      -   280K  -
hrpool/home/reports@thursday       0      -   538K  -
```

The `listsnapshots` pool property is:

- Used to enable or disable the display of snapshots
- Disabled by default
- Enabled using `zpool set listsnapshot=on` *`<poolname>`*

You use the `zfs list -t snapshot` command to display snapshots as shown in this example.

You can enable or disable the display of snapshot listings in the `zfs list` output by using the `listsnapshots` pool property. This property is disabled by default.

To enable this property, use `zpool set listsnapshots=on` followed by the pool name.

Note: If you disable this property, you must use the `zfs list -t snapshot` command to display snapshot information.

Displaying a ZFS Snapshot

To list snapshots created for a specific file system, enter `zfs list -r -t snapshot` followed by the file system name.

```
# zfs list -r -t snapshot -o name,creation hrpool/home
NAME                              CREATION
hrpool/home/reports@tuesday       Tue Nov 29 10:08 2011
hrpool/home/reports@wednesday     Wed Nov 30 08:05 2011
hrpool/home/reports@thursday      Thu Dec  1 07:03 2011
hrpool/home/bonus@now             Fri Dec  2 06:15 2011
```


You can list snapshots that were created for a particular file system by using the `zfs list` command with the `-r` and `-t snapshot` options, followed by the file system name.

In the example shown here, you have decided to list the snapshots created for the file system `hrpool/home`. You have chosen to display this information by using the `name` and `creation` properties.

Note: These are only sample options. For more information about the options you can use with the `zfs list -t snapshot` subcommand, see the `zfs(1)` man page.

Snapshot Space Accounting

- When a snapshot is created, its space is:
  - Initially shared between the snapshot and the file system
  - Possibly shared with previous snapshots
- As the file system changes, previously shared space:
  - Becomes unique to the snapshot
  - Is counted in the snapshot's `used` property
- Deleting snapshots can increase the amount of space that is unique to (and thus used by) other snapshots.

Note: A snapshot's space `referenced` property is the same as the file system's was when the snapshot was created.

Snapshot Space Accounting

To display how much space is consumed by snapshots and descendent file systems, use `zfs list -o space`.

```
$ zfs list -o space
NAME               AVAIL   USED  USEDSNAP  USEDDS   USEDREFRESERV   USEDCHILD
rpool              25.4G  7.79G         0  64K                  0       7.79G
rpool/ROOT         25.4G  6.29G         0  18K                  0       6.29G
rpool/ROOT/sol11   25.4G  6.29G         0  6.29G                0           0
rpool/dump         25.4G  1.00G         0  1.00G                0           0
rpool/export       25.4G    38K         0   20K                 0         18K
rpool/export/home  25.4G    18K         0   18K                 0           0
rpool/swap         25.8G   512M         0  111M              401M           0
```


From the output shown in this example, we can see how much space is available in each file system, how much space is being used, how much space is being consumed by snapshots of each dataset (`USEDSNAP`), how much space is being used by the dataset itself (`USEDDS`), how much space is being used by a refreservation set on the dataset (`USED REFRESERV`), and how much space is being used by the children of this dataset (`USEDCHILD`).

Now, you look at how to create and destroy clones.

Creating a ZFS Clone

To create a clone, enter `zfs clone` followed by the snapshot name from which the clone is to be created, and the name of the new file system or volume.

```
# zfs snapshot hrpool/ws/gate@yesterday
# zfs clone hrpool/ws/gate@yesterday hrpool/home/reports/bug123
```

The new file system or volume:

- Can be located anywhere in the ZFS hierarchy
- Has the same dataset type (for example, file system or volume) as the snapshot from which the clone was created

Note: A clone of a file system must be created in the same pool where the original file system snapshot resides.

ORACLE

In the example in the slide, create a clone named `hrpool/home/reports/bug123` that is being created with the same initial contents as the snapshot `hrpool/ws/gate@yesterday`. Because you can write to a clone, you can now use this clone for testing purposes (so that you do not affect the production version of the file system) or as a way to modify the production system by making the changes to the clone and then overlaying the production system.

Relationship of Clone and Snapshot

- A clone can be created only from a snapshot.
- An implicit dependency exists between the clone and snapshot.
- The original snapshot cannot be destroyed as long as the clone exists.
  - The `origin` property exposes this dependency.
  - The `zfs destroy` command lists any such dependencies (if they exist).
- A clone does not inherit the dataset properties of the dataset from which it was created.

Note: Use the `zfs get` and `zfs set` commands to view and change the properties of a cloned dataset.

Destroying a ZFS Clone

To destroy a clone, use `zfs destroy` followed by the clone name.

```
# zfs destroy hrpool/home/reports/bug123
```


In the example shown here, you are destroying the clone named `hrpool/home/reports/bug123`.

Remember that clones must be destroyed before the parent snapshot can be destroyed.

Quiz

You want to create a snapshot named `thursday` of the file system `/hrpool/home/smith`. Which of the following commands would you use to do this?

a. `zfs snapshot thursday hrpool/home/smith`
b. `zfs snapshot hrpool/home/smith thursday`
c. `zfs snapshot hrpool/home/smith@thursday`
d. `zfs snapshot hrpool/home/smith_thursday`

Answer: c

Practices 5-3 and Practice 5-4 Overview: Administering ZFS Snapshots and Clones and Administering ZFS Pools with Disk Slices

These practices cover the following topics:

- Creating a ZFS snapshot
- Destroying a ZFS snapshot
- Creating a ZFS clone
- Destroying a ZFS clone
- Creating a ZFS storage pool with disk slices

ORACLE

These two practices should take you a total of approximately 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for data storage management
- Administer ZFS storage pools
- Administer ZFS file systems
- Administer ZFS snapshots and clones

In this lesson, you learned how to set up and administer data storage by using ZFS storage pools, file systems, snapshots, and clones.