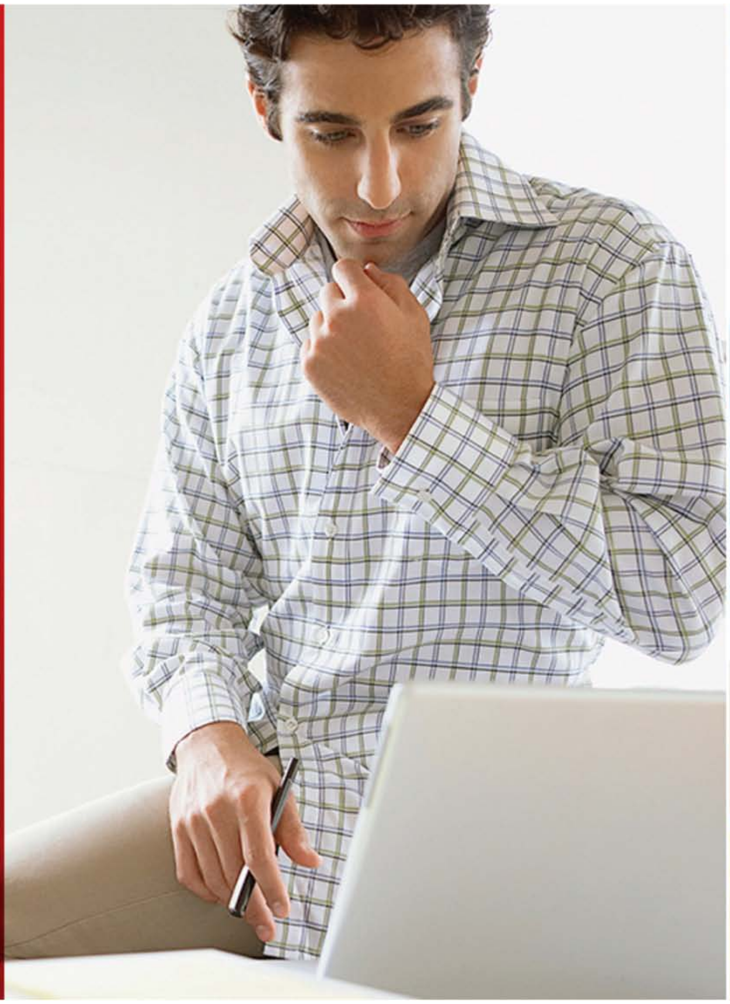
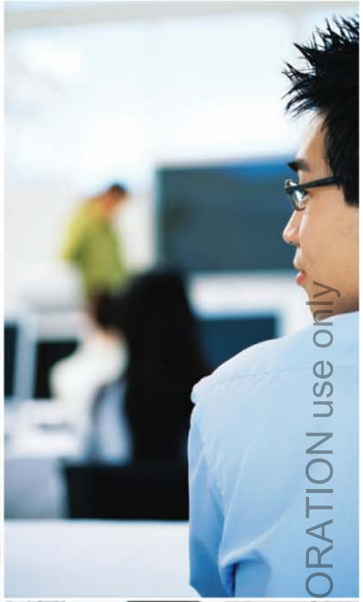
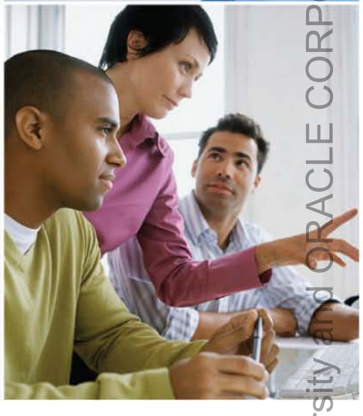
ORACLE
UNIVERSITY

Hardware and Software
Engineered to Work Together







Oracle Solaris 11 System Administration

Student Guide – Volume I

D72896GC40

Edition 4.0 | September 2014 | D88132

Learn more from Oracle University at **oracle.com/education/**

ORACLE

Author

Vijetha M Malkai

Technical Contributors and Reviewers

Muhammad Aseel Khan

Rajesh Rajasekharan

Gary Riseborough

David Maxwell

Editors

Vijayalakshmi Narasimhan

Smita Kommini

Graphic Designers

Maheshwari Krishnamurthy

James Hans

Publishers

Nita Brozowski

Syed Imtiaz Ali

Contents

1

Introduction

Overview

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

Course Goals

The goals of this course are to:

- 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
- 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 by using the operating system.

The goals of this course are to:

- Provide you with the skills that 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 your job.
- Present tasks that cover the full range of system administrative responsibilities, including installing the OS, managing software packages and services, and administering networks, data storage, zones, and users
- 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 the Oracle Solaris 11 Operating System
- Lesson 3: Managing Boot and Shutdown of a System
- Lesson 4: Administering Services by Using SMF

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.

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

- Lesson 2 titled "Installing the Oracle Solaris 11 Operating System" covers installation of the Oracle Solaris 11 operating system by using an interactive installer. You learn about planning for the installation, and then performing the installation. You also learn to verify the OS installation and to establish a system baseline. As part of lesson 2, you practice performing an interactive installation, as well as verification of the installation.
- Lesson 3 titled "Managing the Boot and Shutdown of a System" covers the boot design and the boot and shutdown process of x86 and SPARC systems.
- Lesson 4 titled "Administering Services by Using SMF" is about administering services, which introduces you to the Service Management Facility (SMF) feature and its components. You also learn to administer SMF services and manage SMF services by using the SMF GUI.

Course Agenda: Day 2

- Lesson 4: Administering Services by Using SMF (continued)
- Lesson 5: Administering Software Packages by Using IPS
- Lesson 6: Managing Data by Using ZFS

- On Day 2, you complete learning about the SMF-related concepts in lesson 4 that was started on Day 1 and then move to lesson 5.
- Lesson 5 titled "Administering Software Packages by Using IPS" discusses IPS, its components, and interfaces. You also learn how to configure an IPS client to access the local IPS repository, manage package publishers, manage software packages, and manage signed packages and package properties.
- Lesson 6 titled "Managing Data by Using ZFS" covers setting up and administering data. You start with understanding the role of ZFS in data management and continue with learning about administering ZFS storage pools, ZFS file systems, ZFS properties, and ZFS snapshots and clones.

Course Agenda: Day 3

- Lesson 6: Managing Data by Using ZFS (continued)
- Lesson 7: Administering the Network

- On Day 3, you continue with lesson 6 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 data management.
- In lesson 7 titled "Administering the Network," you learn about some of the basic networking concepts. You also learn to administer a datalink configuration, network interface, and profile-based network configuration, configure a virtual network, verify network operations, and manage resources on the network.

Course Agenda: Day 4

- Lesson 8: Administering Oracle Solaris Zones
- Lesson 9: Controlling Access to Systems and Files

- Lesson 8 titled “Administering Oracle Solaris Zones” is about administering zones, and covers features of Oracle Solaris Zones, how to configure an Oracle Solaris Zone, and how to determine an Oracle Solaris Zone configuration.
- Lesson 9 titled “Controlling Access to Systems and Files” is about controlling access to systems and files. You begin by learning 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)
- Lesson 10: Administering User Accounts
- Lesson 11: Managing System Processes and Scheduling System Tasks

- You continue with 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 titled "Administering User Accounts," you learn to plan for user administration, to set up and maintain user accounts, to manage user initialization files, and to configure user shells and user disk quotas.
- In lesson 11 titled "Managing System Processes and Scheduling System Tasks," you look at how to manage system processes and schedule system tasks.

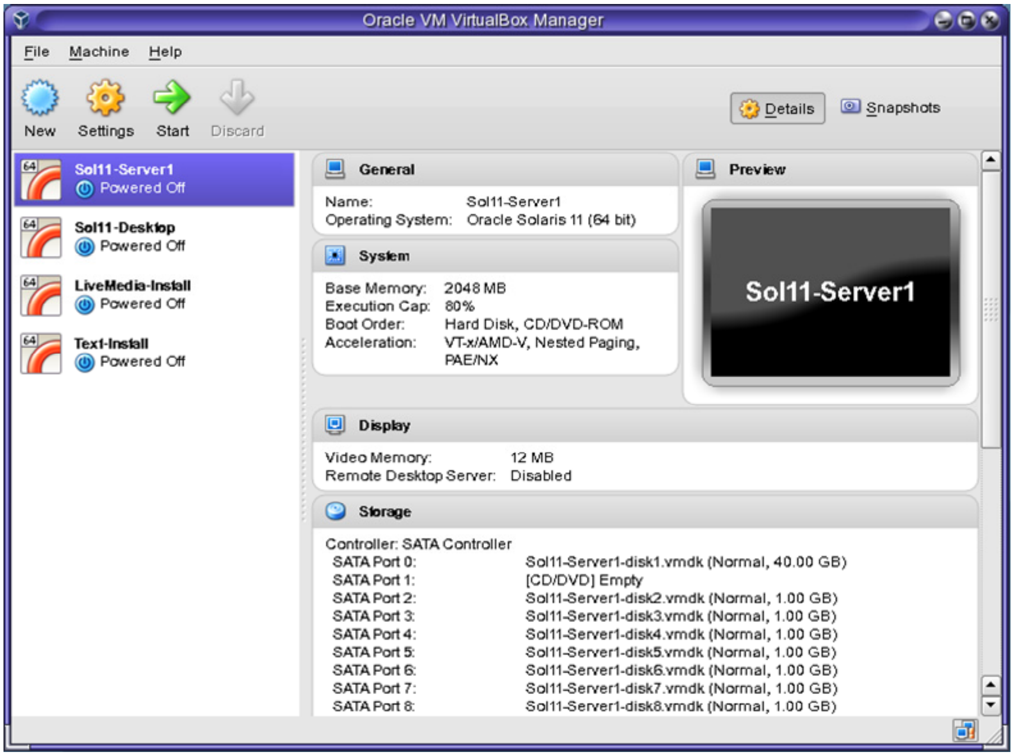
Introductions

- Name
- Company affiliation
- Title, function, and job responsibility
- Experience related to the 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 learned in the lecture. The lab environment that you 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 this lesson. 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 the Oracle Solaris 11 Operating System

Objectives

After completing this lesson, you should be able to:

- Describe Oracle Solaris 11 OS
- Implement a plan for an Oracle Solaris 11 OS installation
- Install the Oracle Solaris 11 OS by using the Live Media installer
- Install the Oracle Solaris 11 OS by using the text installer
- Verify the installed OS

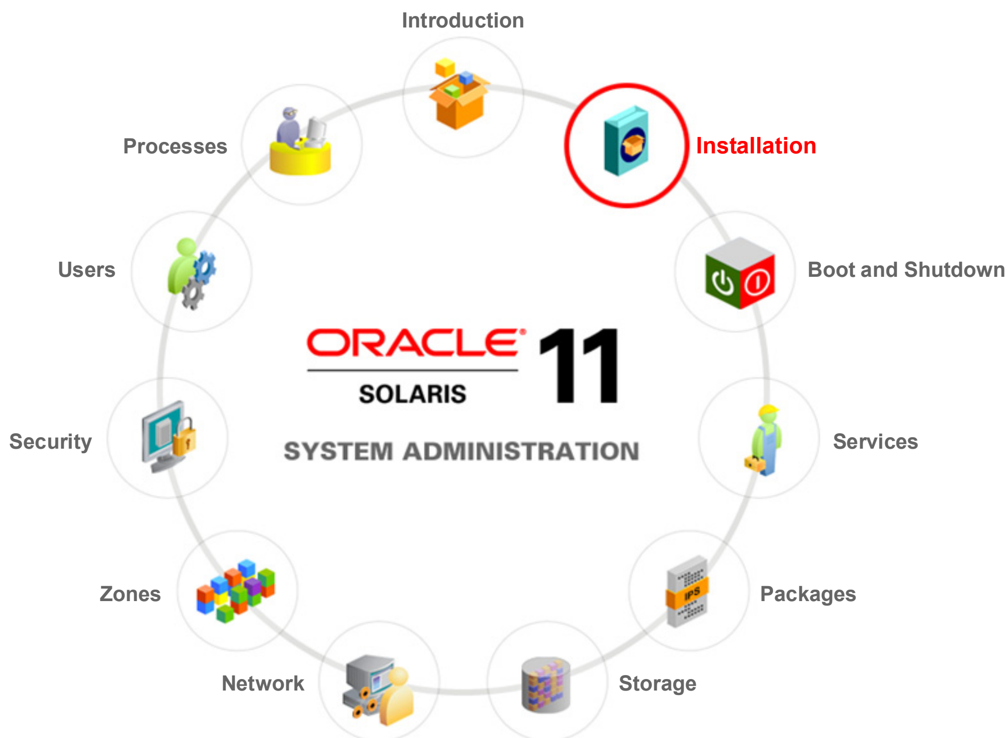
This lesson begins with a brief introduction of the Oracle Solaris 11 operating system, and goes on to help you to identify the list of tasks to be performed before installing the Oracle Solaris 11 OS.

Next, you learn how to install Oracle Solaris 11 by using the Live Media and text installers.

The lesson concludes by discussing how to verify the OS 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 in the diagram). You learn about why certain tasks precede or follow other tasks, and the importance of each task as it pertains to a system administrator's job. As indicated in the diagram, you start with the OS installation, followed by the boot and shutdown process, Service Management Facility (SMF), and so on.

In this lesson, you are taught about the various methods of installing the Oracle Solaris 11 OS and how to perform interactive installations by using the text installer and Live Media ISOs.

Lesson Agenda

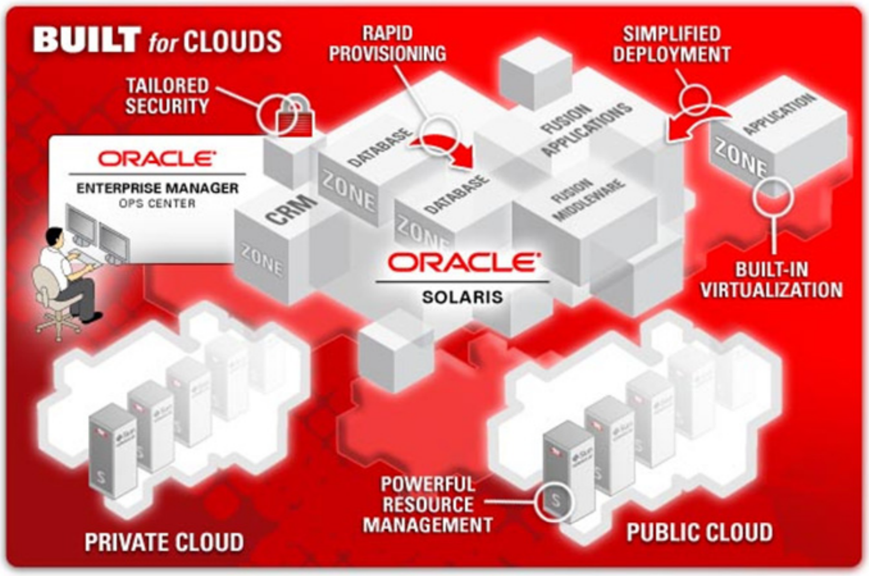
- **Introduction to Oracle Solaris 11 OS**
- Planning for an Oracle Solaris 11 OS Installation
- Installing Oracle Solaris 11 OS by Using the Live Media Installer
- Installing Oracle Solaris 11 OS by Using the Text Installer
- Verifying the OS Installation

Introduction to Oracle Solaris 11 OS

- Is supported by over 11,000 applications
- Provides a complete, reliable, secure, and simple solution for deploying your enterprise-grade clouds
- Provides centralized cloud management with complete OpenStack distribution
- Delivers unique features to increase performance, streamline management, and automate support for Oracle deployments

- Oracle Solaris 11 has the largest installed base of any enterprise UNIX operating system.
- It runs on more than thousand systems from leading manufacturers, and holds hundreds of world records on Oracle's Sun x86-based X-series servers and on SPARC-based T-series and M-series servers.
- It is co-engineered with Oracle's software and hardware to run Oracle's enterprise applications, scalable systems, high-performance interconnects, and optimized data center storage to achieve highest performance and efficiency.
- It optimizes the system CPU, memory, I/O, and network resources for a broad range of mission-critical workloads.
- No longer simply an OS, the features and enhancements included in Oracle Solaris 11 deliver no-compromise virtualization, application-driven software-defined networking, and a complete OpenStack distribution for creating an enterprise cloud that allows you meet IT demands and re-define your business.
- It outperforms other operating systems in a range of key enterprise benchmarks, including OLTP (TPC-C), BIEE (Oracle BIEE), SAP ERP (SAP SD 2-tier), Web (SPECweb), Mail (SPECmail), PeopleSoft Payroll, and Customer Relationship Management (CRM).

Key Benefits of Oracle Solaris 11



Key Benefits of Oracle Solaris 11

- Simplified administration
- Built-in virtualization
- Scalable data management
- Advanced protection
- Zero-to-complete cloud in minutes with Oracle Solaris and OpenStack
- Greater flexibility with independent and isolated Kernel Zones virtualization
- Fast and agile application provisioning with Unified Archives
- Conformation with service-level agreements by using application-driven software-defined networking
- Risk reduction with comprehensive compliance checking and reporting

The Oracle Solaris 11 OS provides several new features to simplify administration, enable rapid application provisioning through designed-in virtualization, ensure data integrity with scalable data management, and provide advanced protection with the highest-level of security. It delivers ground-breaking features for secure and fast deployment of services in large-scale cloud environments, as well as enterprise data center environments.

- **Built-in virtualization:** Oracle Solaris 11 provides a solution with the lowest overhead and latency in the market. Software-defined networking drives network virtualization into the network infrastructure. The OS also provides secure live migration with Oracle VM for SPARC and x86.
- **Scalable data management:** Oracle Solaris ZFS, which is the default file system in Oracle Solaris 11, brings advanced storage features such as built-in deduplication, encryption, and thin provisioning to enterprise servers. It provides the Oracle Solaris ZFS snapshots feature that is tightly integrated with the IPS software packaging systems to deliver boot environments that allow fail-safe updates. These snapshots have very low overhead.

- **Advanced protection:** Oracle Solaris 11 provides fully integrated security for users, applications, and devices while simplifying administration with fine-grained delegated management, implementing the latest security standards, and continuing Oracle Solaris' leadership position as a highly secure operating system.
- **Centralized cloud management with OpenStack:** Oracle Solaris 11 provides complete OpenStack distribution. OpenStack, the popular open source cloud computing software that is enjoying widespread industry involvement, provides comprehensive self-service environments for sharing and managing compute, network, and storage resources in the data center through a centralized web-based portal. Integrated into all the core technology foundations of Oracle Solaris 11, you can now set up an enterprise-ready, private cloud Infrastructure-as-a-Service (IaaS) environment in minutes. For more information about OpenStack for the Oracle Solaris 11 OS, refer to http://www.oracle.com/technetwork/server-storage/solaris11/technologies/openstack-2135773.html.
- **Greater flexibility with independent and isolated Kernel Zones virtualization:** Oracle Solaris Zones have been enhanced to include support for fully independent and isolated environments, called Oracle Solaris Kernel Zones, that provide a full kernel and user environment within a zone. Kernel Zones increase operational flexibility, and are ideal for multitenant environments where maintenance windows are significantly harder to schedule.
- **Fast and agile application provisioning with Unified Archives:** The Oracle Solaris 11 OS introduces Unified Archives, a new type of archive format that enables developers and administrators to quickly clone or create a complete backup of their environments for fast provisioning in the cloud or in the case of disaster recovery. You can quickly capture a complete bare metal system, virtual environments, or a combination of both.
- **Application-driven software-defined networking:** A new socket-level flow API allows an application to directly prioritize its traffic through a series of network flows, which leads to optimized application performance, and to reduce any adverse impact of resource contention. This application-driven software-defined networking, from application through storage, along with administrative-driven flows, help to ensure that service-level agreements are maintained within a data center or cloud environment.
- **Risk reduction with comprehensive compliance checking and reporting:** Businesses can meet their compliance requirements by using a new `compliance`(1M) tool that manages a variety of compliance benchmarks and assessments. It provides a standardized approach to maintaining the security of enterprise systems, such as:
  - Automatically verifying the presence of critical updates
  - Checking system security configuration settings
  - Examining systems for signs of compromise

Platforms Supported by Oracle Solaris 11 OS

Architecture	Systems	Virtualization	OS Virtualization
SPARC	M6-32 M5-32	Dynamic Domains, Logical Domains	Oracle Solaris Zones
	T-Series	Oracle VM for SPARC formerly known as LDoms	
x86	X86 (64-bit processor)	Oracle VM for x86	

Note: Third-party virtualization offerings from vendors, including VMware, Windows, and Red Hat, are also supported.

Oracle Solaris 11 supports a range of architectures and virtualization layers. The table in the slide summarizes various platforms supported by Oracle Solaris 11. For more information, refer to the hardware compatibility list (HCL) that is available at http://www.oracle.com/webfolder/technetwork/hcl/data/s11ga/index.html.

Integration of Oracle Solaris 11 with the Oracle Stack

- Record performance in running Oracle Database, Oracle Middleware, and Oracle Applications
- New high-performance, super scalable virtual memory
- Reduced down time with new Optimized Shared Memory interface
- Kernel Mode Acceleration for Oracle Real Application Clusters (Oracle RAC)
- Faster transparent hardware cryptography acceleration
- Unique observability with Oracle Solaris DTrace
- High availability and disaster recovery across the Oracle Stack
- Integrated development environment

Enterprises increasingly need to manage cost, reduce time to market, and improve business agility. They need to ensure security, scalability, and reliability for deployments and simplify the management of and support for their full solution stack, which most often includes Oracle databases, Oracle applications, and Oracle middleware. Oracle Solaris 11 is uniquely positioned to address these needs across the Oracle stack. Oracle Solaris is co-engineered with the Oracle stack to integrate Oracle's enterprise applications, scalable systems, high-performance interconnects, and optimized data center storage.

Engineered together with Oracle hardware, Oracle database, middleware, and applications, Oracle Solaris 11 delivers unique features to increase performance, streamline management, and automate support for Oracle deployments.

- **World-record performance:** Oracle software, which runs on the combination of Oracle hardware and Oracle Solaris, can leverage many system-level enhancements to deliver optimized performance. Oracle Solaris 11 is architected to provide support for tens of thousands of CPU cores and hundreds of terabytes of memory capacity, practically limitless virtualization, and even higher reliability, availability, and serviceability (RAS) for future SPARC and x86 servers.

- **New high-performance, super scalable virtual memory subsystem:** Oracle Solaris 11 is designed with a new virtual memory system that can not only scale but also optimize the assignment of memory resources. The result of this designing is a high-performance, virtual memory system that scales with the size of the system memory. Very large memory pages are now supported with the Oracle Solaris 11.1 virtual memory system that transparently speeds applications such as databases and enterprise applications.
- **Reduced down time:** Oracle Solaris now includes a new Optimized Shared Memory (OSM) interface, which is a dynamic, NUMA-optimized, granular shared memory that offers flexibility without compromising performance or functionality. OSM enables dynamic resizing of the Oracle Database System Global Area (SGA) without having to reserve memory and reboot the Oracle Database, unlike other operating systems such as IBM AIX or Red Hat Linux. It also allows faster startup of Oracle Database instances by using fewer OSM segments, which can be increased later as required. The next generation Oracle Database technology is scheduled to use OSM to allow online resizing of the SGA and avoid down time.
- **Oracle RAC Kernel Mode Acceleration:** The new Oracle RAC Kernel Mode Acceleration in Oracle Solaris 11.1 filters the database block requests that are destined for the Oracle RAC Lock Manager Service (LMS) processes. For more information, refer to http://www.oracle.com/technetwork/articles/servers-storage-admin/sol-why-os-matters-1961737.html.
- **Software in Silicon:** Oracle has the unique ability to add software instructions that will accelerate Oracle software into the SPARC processor, thus making the combination of Oracle software on Oracle hardware the optimal choice. An example of the Software in Silicon technology is the offloading of database cryptography tasks on to the SPARC T4 and T5 chips that makes encrypting database tables a task with the least overhead. The Oracle Solaris cryptographic framework transparently utilizes SPARC cryptographic accelerators, producing extremely high-performance encryption with no application code changes or additional configurations.
- **Vital Observability:** Oracle Solaris DTrace is a comprehensive dynamic tracing facility (built into Oracle Solaris) that is available for administrators and developers on live production systems to examine the behavior of both user programs and of the OS itself. Oracle Solaris DTrace enables you to explore your system to track down performance problems across many layers of software, or to locate the cause of aberrant behavior.
- **High Availability and Disaster Recovery Across the Stack:** Integrated with Oracle Solaris at the kernel level, Oracle Solaris Cluster enables high availability and disaster recovery for Oracle databases and applications. The solutions are seamlessly integrated and stringently tested together with a spectrum of Oracle Sun hardware. It is the HA component for Oracle Optimized Solutions and SPARC SuperCluster. Oracle Solaris Cluster also supports Oracle Solaris Zones and Oracle VM Server for SPARC, and is configurable to offer protection at the application, zone, or VM level with policy-based recovery behavior and reliable management of multi-tier dependencies for virtualized deployments.
- **Integrated Development Environment:** Oracle database application development is streamlined with the Oracle Solaris Studio 12.3 IDE, which supports Pro*C and OCI-based applications, simplified setup with database project and connection wizards, and the ability to manage database schemas, tables, and data within the IDE. Oracle Solaris Studio 12.3, Oracle's advanced C, C++, and Fortran development tool suite deliver the latest in compiler optimizations, multithread performance, and powerful analysis tools for optimal application performance and reliability on Oracle Solaris. It includes a Performance Analyzer that seamlessly combines and reconciles two call stacks (the Java stack and the machine stack) into a single view.

Lesson Agenda

- Introduction to Oracle Solaris 11 OS
- **Planning for an Oracle Solaris 11 OS Installation**
- Installing Oracle Solaris 11 OS by Using the Live Media Installer
- Installing Oracle Solaris 11 OS by Using the Text Installer
- Verifying the OS Installation

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 questions such as:
  - How many users will you need to support?
  - What applications will you be running?
  - What type of network will you be using?
  - What are your data storage needs?
  - What are your hardware needs?

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. Because the OS is the backbone of data center operations, and must be able to support the needs of the business, it is important that it is installed and configured properly. This requires planning.

Methods of Installing Oracle Solaris 11 Operating System

	Interactive		Automated
	Live Media	Text Installer	Automated Installer (AI)
SPARC		✓	✓
x86	✓	✓	✓
Single system	✓	✓	✓
Multiple-client systems			✓

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

- Live Media for x86-based systems
- Text installer, which can be used on both x86 and 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 “hands-free” network installation on a single system or on multiple client systems. The automated installation option 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 addition to the installation options already described, you have the following options for installing and modifying Oracle Solaris 11 OS:

- **Create custom installation images:** You can create a preconfigured Oracle Solaris installation image by using the distribution constructor tool. The tool takes a customized XML manifest file as input and builds an installation image that is based on the parameters specified in the manifest file. You can build a custom image based on any of the default installation images. For example, you could build a custom text installer image or a custom GUI installer image.

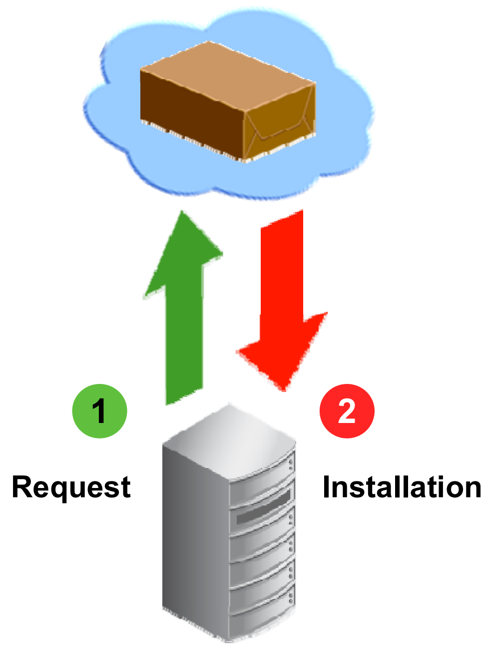
- **Update an installed Oracle Solaris system:** You cannot use an installer to update an existing Oracle Solaris system. Instead, you need to use the `pkg` utility to access the package repositories and download new or updated software packages for your system.

In this course, you learn mainly about performing an interactive installation. Installing using AI, creating customized installation images, and updating an installed Oracle Solaris 11 OS are covered in depth in the *Oracle Solaris 11 Advanced System Administration* course, which is the follow-on to this course.

Differences Between Live Media and Text Installer

Feature	Live Media	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	Always configures `root` as a role	The `root` might or might not be a role.
Memory	Requires more memory than the text installer	Requires less memory than the Live Media GUI installer

Installation Process



Oracle Solaris 11 OS download website:
http://www.oracle.com/technetwork/server-storage/solaris11/downloads

To prepare for the installation, you first download and save the ISO image on your system. The steps to download the OS are as follows:

1. Visit the Oracle Solaris 11 download website (the site's URL is provided in the slide).
2. Accept the license agreement.
3. Click the ISO image that you want to download.
4. Sign-in to your single sign-on (SSO) account.
5. Save the ISO image on your system.

After the ISO is downloaded, perform the pre-installation tasks, and then install the OS.

After you download the image, you can copy it to removable media such as a CD, DVD, or USB flash drive.

Note: If you want to burn the image to a USB flash drive, download a USB image. For USB images, you need the `usbcopy` utility in order to copy the image to a USB flash drive. You can add this utility to your system by installing the `pkg:/install/distribution-constructor` package.

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 pre-installation tasks, such as identifying system requirements and additional installation considerations for running the installer on your system, and ensuring that you have all 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 Oracle Solaris documentation on the Oracle Technology Network (OTN) website under Documentation > Oracle Documentation > Systems Software:

(http://www.oracle.com/technetwork/indexes/documentation/index.html#sys_sw)

It is always a best practice to review the documentation—specifically the release notes—as part of your pre-installation tasks.

Identifying System Requirements

Installer	Package Group	Memory	Recommended Minimum Disk Space
Live Media	`solaris-desktop`	2 GB	13 GB
Text installer	`solaris-large-server`		9 GB

Notes

- Oracle Solaris 11 runs on both x86 (64-bit only) and SPARC hardware.
- The text installer requires less memory than the Live Media installer. The exact minimum requirement varies depending on system specifications. If your system does not have enough memory to run the GUI installer, use the text installer instead.

Identifying Additional Installation Considerations

- The Live Media ISO image installer is only for 64-bit x86 platforms.
- For SPARC-based systems, use the Text or Automated Installer.
- Interactive installers can perform an initial installation on:
  - The whole disk
  - The Oracle Solaris x86 partition
  - The SPARC slice (text installer)
- (Caution): The installation overwrites all the existing data on the targeted disk.

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

The OS 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.

Note: As observed in your practice environment, you can install VirtualBox (an x86-based virtualization software) on your machine and learn to use Oracle Solaris 11 OS by installing it on a VirtualBox VM.

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

Checking Device Drivers

- Device drivers enable communication between the operating system and the system's devices.
- Take a few minutes to verify that your system has the appropriate drivers required to manage each of its devices.
  - Before or after an OS installation, use the Oracle Device Detection Tool to determine whether a device driver is available.
  - After the OS installation, use Oracle Device Driver Utility (DDU) to obtain information about devices and their drivers.

Before installing the Oracle Solaris 11 OS, you need to determine whether your system's devices are supported. Review the Hardware Compatibility Lists (HCLs) at http://www.oracle.com/webfolder/technetwork/hcl/index.html. The HCLs provide information about the hardware that is certified or reported to work with the Oracle Solaris operating system.

You can also use the Oracle Device Detection Tool before or after an installation to determine whether a device driver is available. The Oracle Device Detection Tool reports whether the current release supports the devices that have been detected on your system. This tool runs on many different systems, including several different Oracle Solaris 11 releases, Windows, Linux, Mac OS X, and FreeBSD. There is a link to the Oracle Device Detection Tool on the HCL (http://www.oracle.com/webfolder/technetwork/hcl/index.html). For instructions on using the Oracle Device Detection Tool, refer to http://docs.oracle.com/cd/E36784_01/html/E36800/sddusing.html#scrolltoc.

After an Oracle Solaris 11 OS installation, you can use the Oracle Device Driver Utility (DDU) to obtain information about the devices on your installed system and the drivers that manage those devices. The DDU reports whether the currently booted operating system has drivers for all the devices that are detected in your system. If a device does not have a driver attached, the Device Driver Utility recommends a driver package to install. For more information about the Device Driver Utility, see http://docs.oracle.com/cd/E36784_01/html/E36800/useddu.html#scrolltoc.

Lesson Agenda

- Introduction to Oracle Solaris 11 OS
- Planning for an Oracle Solaris 11 OS Installation
- **Installing Oracle Solaris 11 OS by Using the Live Media Installer**
- Installing Oracle Solaris 11 OS by Using the Text Installer
- Verifying the OS Installation

Selecting the Keyboard

```
USB keyboard
 1. Arabic                        15. Korean
 2. Belgian                       16. Latin-American
 3. Brazilian                     17. Norwegian
 4. Canadian-Bilingual            18. Portuguese
 5. Canadian-French               19. Russian
 6. Danish                        20. Spanish
 7. Dutch                         21. Swedish
 8. Dvorak                        22. Swiss-French
 9. Finnish                       23. Swiss-German
10. French                        24. Traditional-Chinese
11. German                        25. TurkishQ
12. Italian                       26. UK-English
13. Japanese-type6                27. US-English
14. Japanese
To select the keyboard layout, enter a number [default 27]:
```


When the GRUB2 menu is displayed, the default entry is automatically used unless you select another option.

The Oracle Solaris 11 Live Media provides a GUI-based interactive installation that guides 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 Live Media with the GUI installer, the first thing that you are asked to do is identify the keyboard layout. The default is US-English [27].

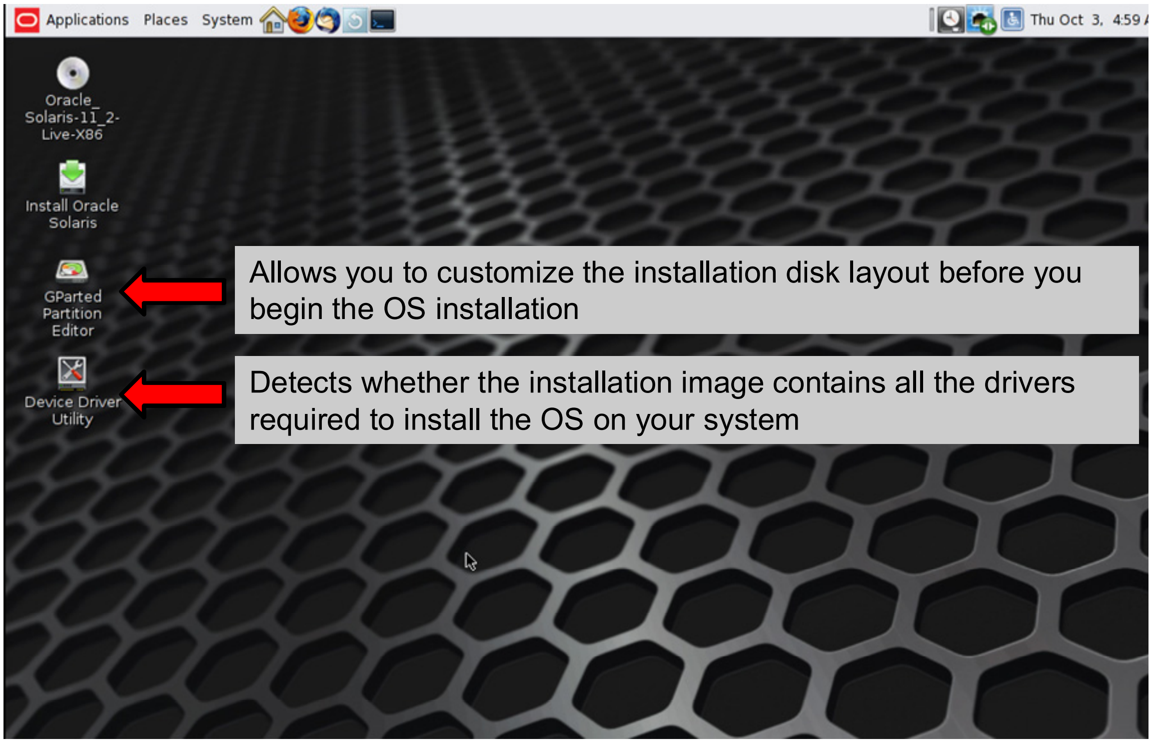
Selecting the Language

```
 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]:
```


After selecting the keyboard layout, you are prompted to select the language that 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.

Note: When booting the Live Media image, the Solaris console login, by default, accepts the username and password as `jack`, and displays the Live Media desktop. This is useful when using Live Media as a troubleshooting tool. If you do log in by using the username and password as `jack`, the default root password is `solaris`.

Introducing the Live Media Desktop



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

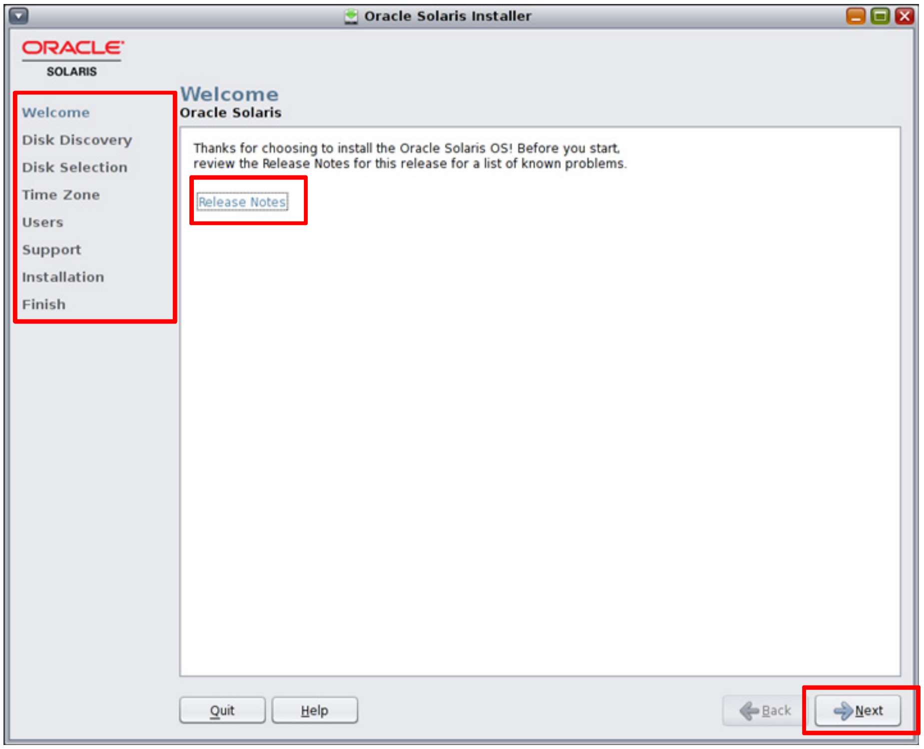
When you boot your system from the Live Media ISO image and the desktop is displayed, if any drivers are missing, a prompt is displayed. Follow the instructions for accessing the Device Driver Utility to locate and install any drivers that are required for the installation.

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

Initiating the Installation with Live Media

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

Welcome Screen



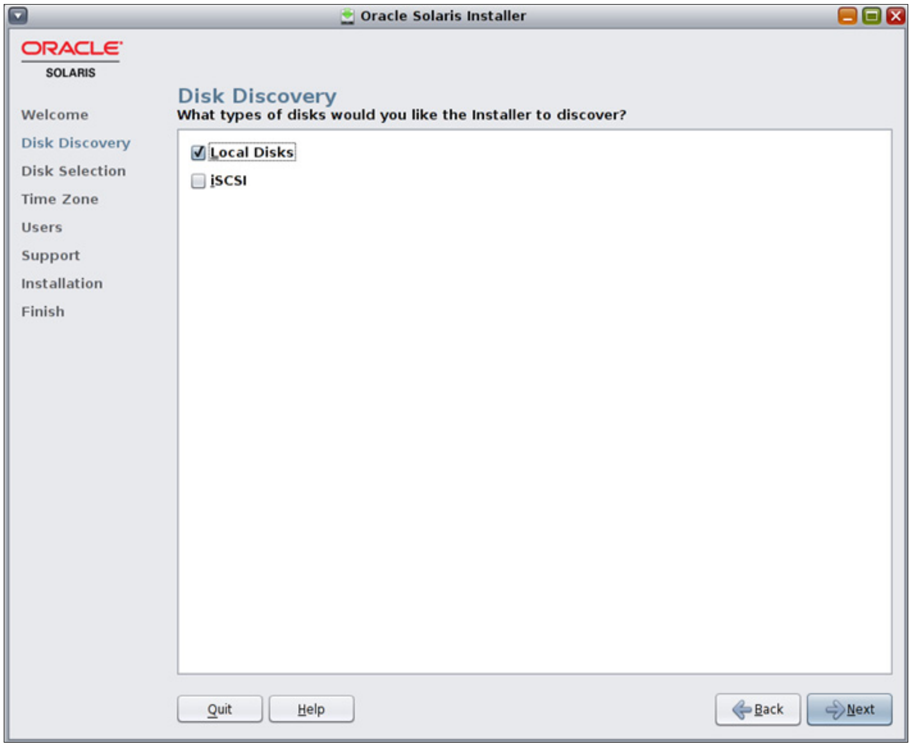
The first screen that you see is the Welcome screen. From this screen, you can review the release notes (if you have not already done so).

The list of items on the left of the Welcome screen highlights the steps that you perform to complete the installation. You begin the installation by providing configuration data for the disk, time zone, and users. The information that you supply during the installation usually should follow your company guidelines.

After you have completed the configuration data screens, the actual installation begins. After the installation is completed, your task is complete.

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

Disk Discovery

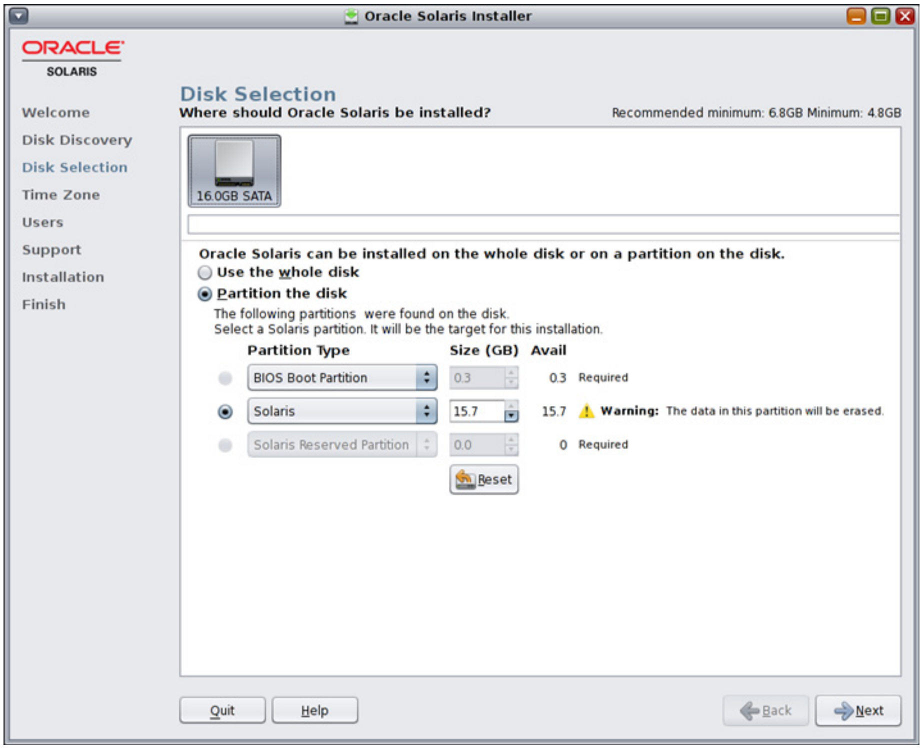


Starting from the Oracle Solaris 11.1 release, the ability to install to iSCSI target LUNs has been included in the LiveCD Installer. Administrators can choose between installing on local disks or iSCSI disks.

- **Local Disks:** This is the default option for disks that are attached to the computer, including internal and external hard disks.
- **iSCSI:** If you want the installer to search for remote disks that are accessible over a network by using the iSCSI standard, select this option. You can connect to a remote iSCSI disk by using DHCP auto-discovery or by manually specifying a target IP address, iSCSI target name and LUN, and initiator name. For more information, refer to the http://docs.oracle.com/cd/E36784_01/html/E36800/guitask.html#scrolltoc guide.

For disk selection, click **Next** to advance to the Disk Selection screen.

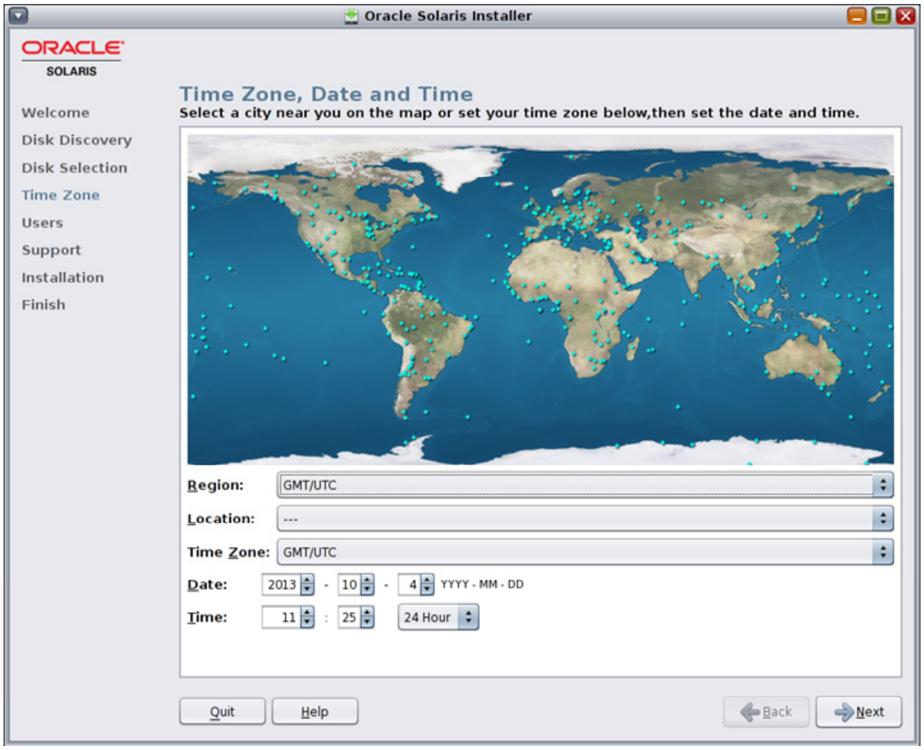
Selecting a Disk



From the Disk Discovery screen, you are taken to the Disk Selection 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.

You have the option to modify the partition layout. For instructions, refer to the guidelines for partitioning a system, which is available at http://docs.oracle.com/cd/E36784_01/html/E36800/guidepart.html#scrolltoc.

Setting the Time Zone, Date, and Time



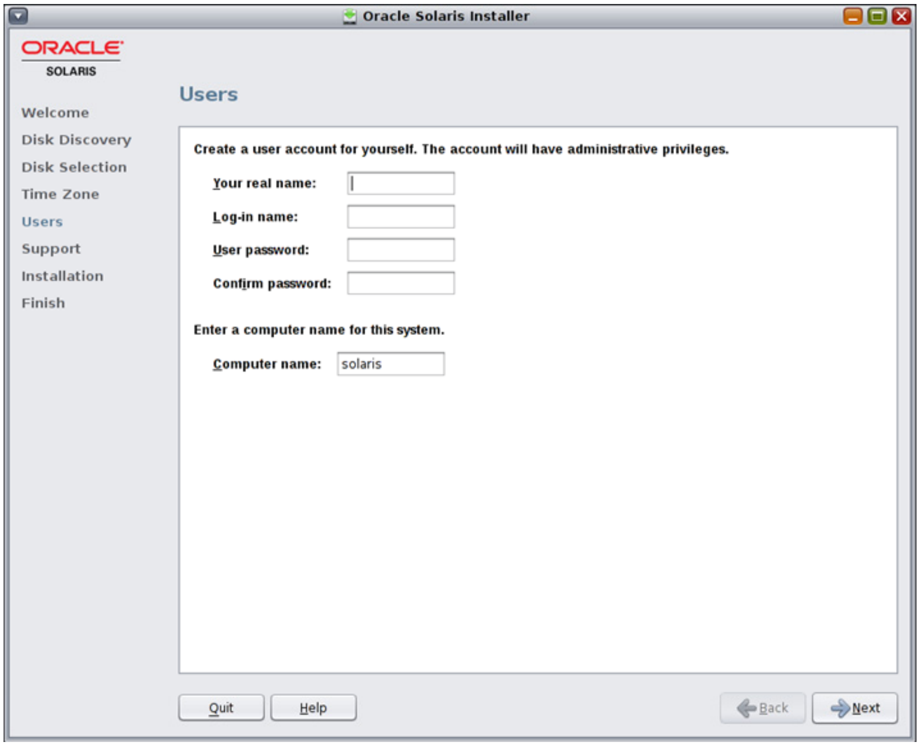
ORACLE

The next screen that appears is the Time Zone, Date and Time screen. 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.

The installer uses the time zone from the system's internal settings as the initial default, if possible. When you select your location on the map, the installer uses that information to set the date, time and time zone.

To continue, click **Next**.

Providing User Information

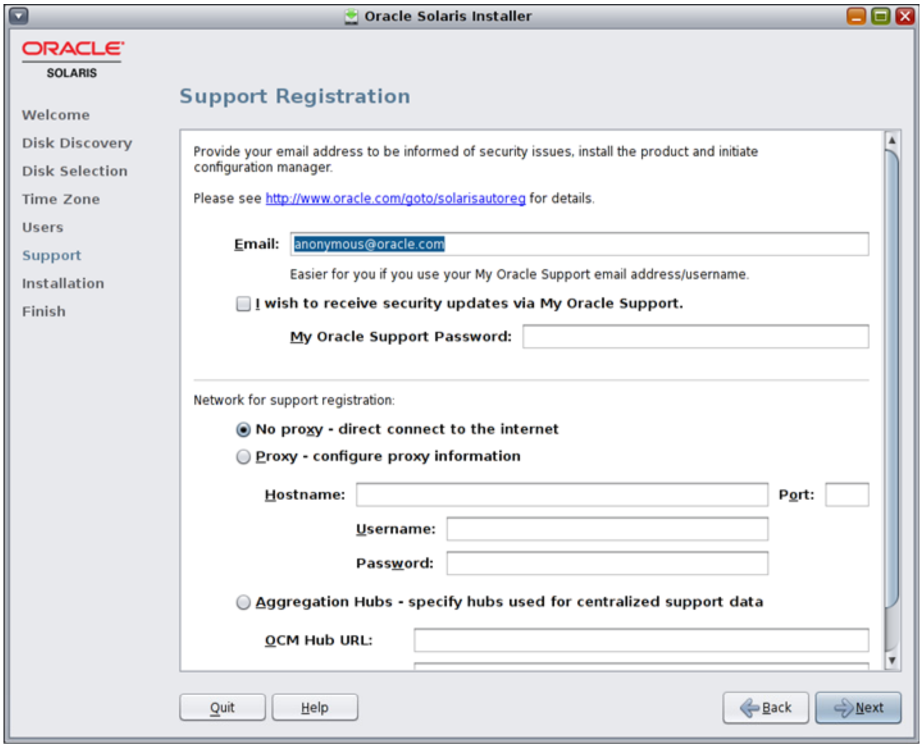


On the Users screen, you enter the user information, including your real name, your login name, and your user password. You are also asked to provide a computer name, which is also referred to as the host name or node name. To continue, click Next.

Notes

- The login name must begin with a letter and can contain only letters and numbers.
- The user account that you create will have administrative privileges (can log in to the `root` role).
- The first time that you use the `root` password, you will be prompted to change it.

Support Registration



ORACLE

The next screen is Support Registration where you can accept the anonymous registration for Oracle Configuration Manager (OCM) and Auto Service Request (ASR). You can also select the option that enables you to receive security updates via My Oracle Support. “No proxy” is selected by default.

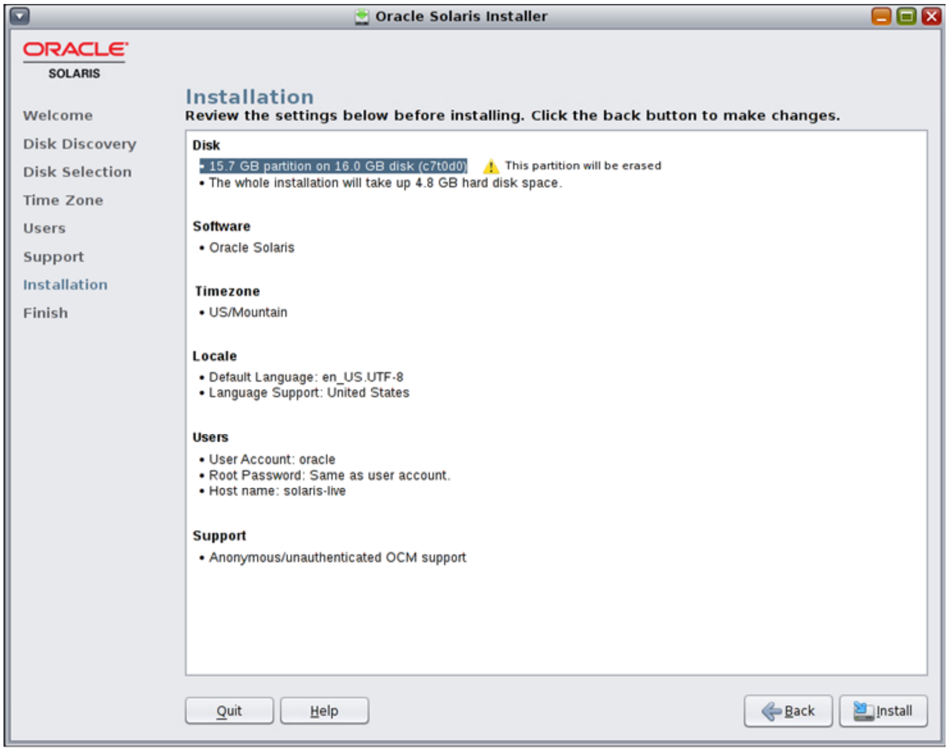
The default Support Registration screen provides an anonymous registration address. If you use this anonymous address with no password, My Oracle Support will receive information about the installed system’s configuration, but will not receive any of your information when the system configuration is uploaded to the Oracle Support organization.

Alternatively, you can register for security updates or disconnect OCM as follows:

- Replace the anonymous email address with your My Oracle Support login ID and password. Use this option if you want to see your information in My Oracle Support and receive security updates. With this option, ASR will also be started.
- Delete the anonymous email address and leave that field blank. OCM will start in a disconnected mode. No data will be sent to My Oracle Support. Or, if you delete the anonymous email address and replace it with another email address other than your MOS login ID, OCM will send data to Oracle Support in an unauthenticated mode.

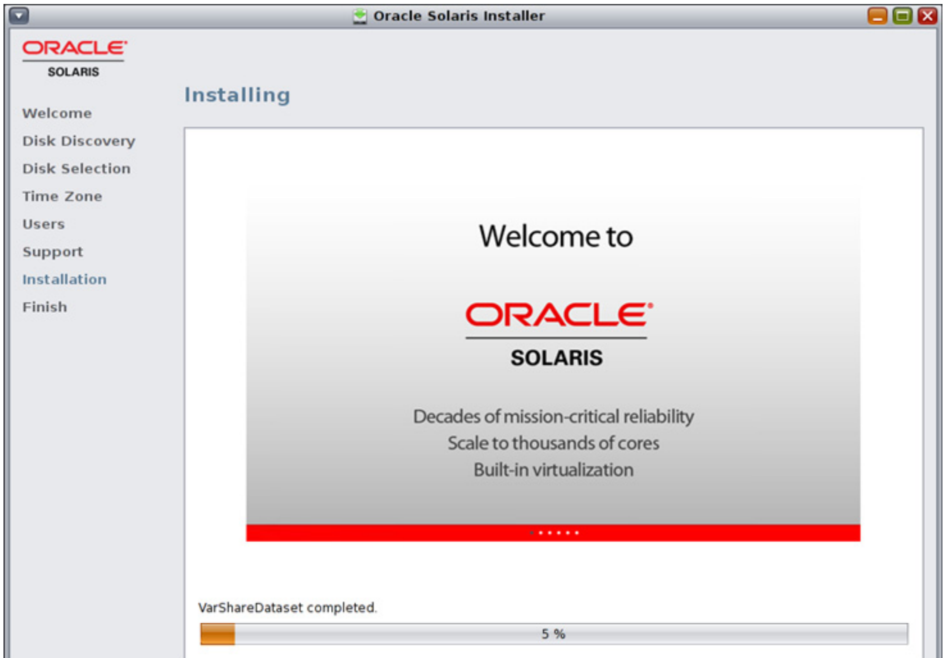
For further information, refer to http://docs.oracle.com/cd/E36784_01/html/E36800/ocm.html#scrolltoc.

Reviewing Installation Specifications



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

Monitoring the Installation



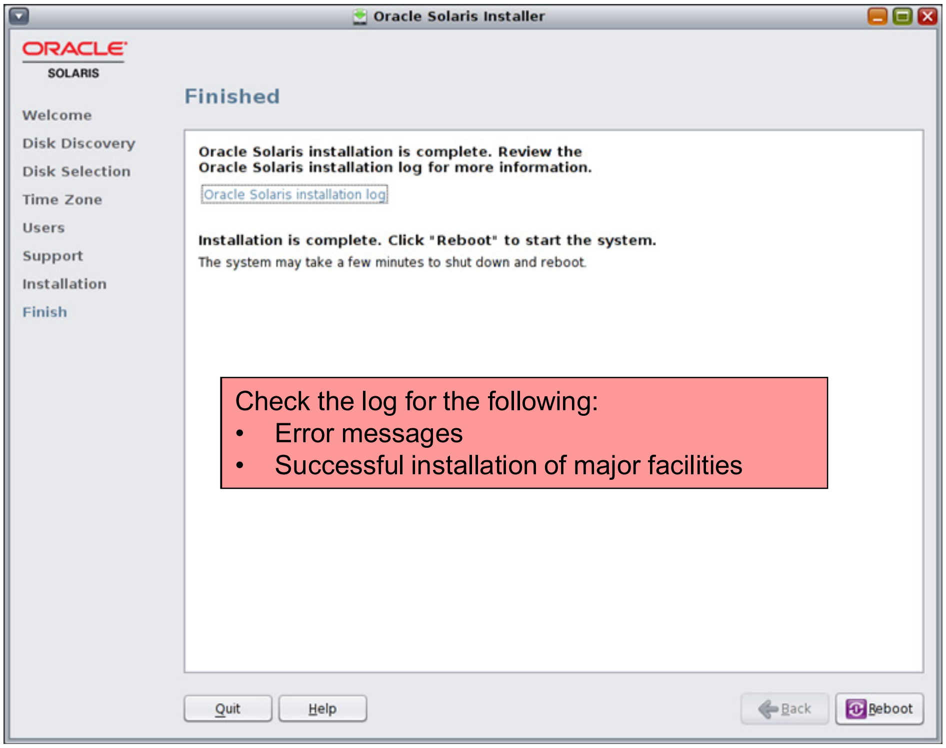
Caution: After the installation starts, do not interrupt it. An incomplete installation can leave a disk in an indeterminate state.

The Installation screen enables you to monitor the progress of the installation. The installation takes about 10 minutes to complete.

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

Note: In your VirtualBox training environment, the installation might take 15 to 20 minutes to complete.

Verifying the Installation

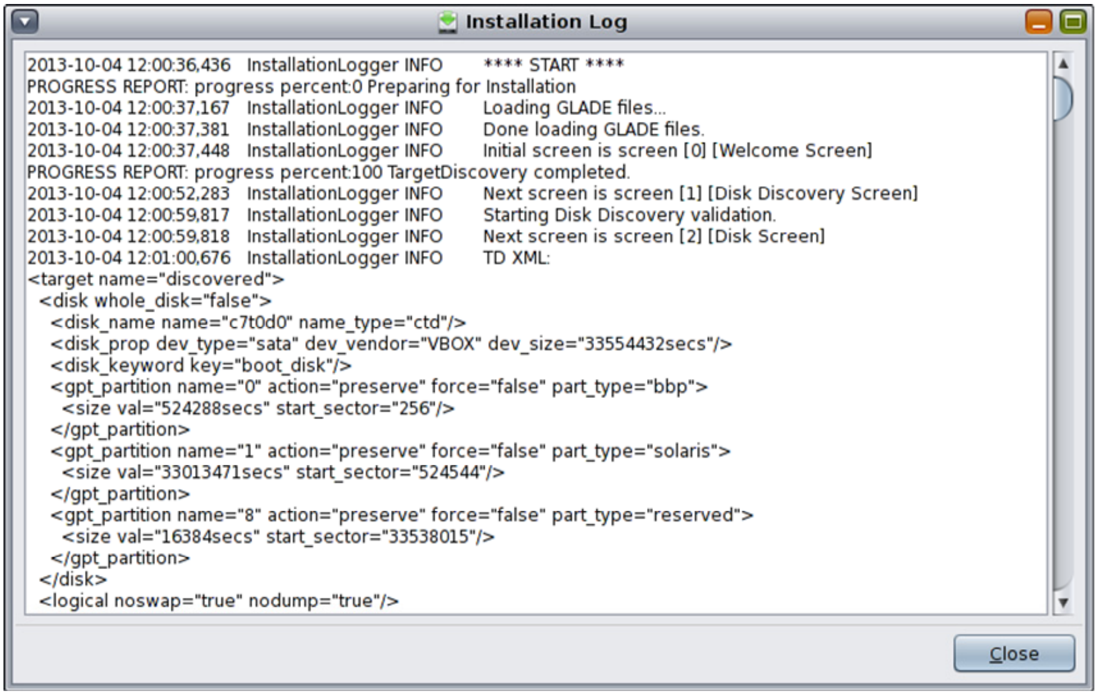


When the installation concludes, you see the Finished screen, which provides you with access to the installation log and an opportunity to verify that:

- 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.

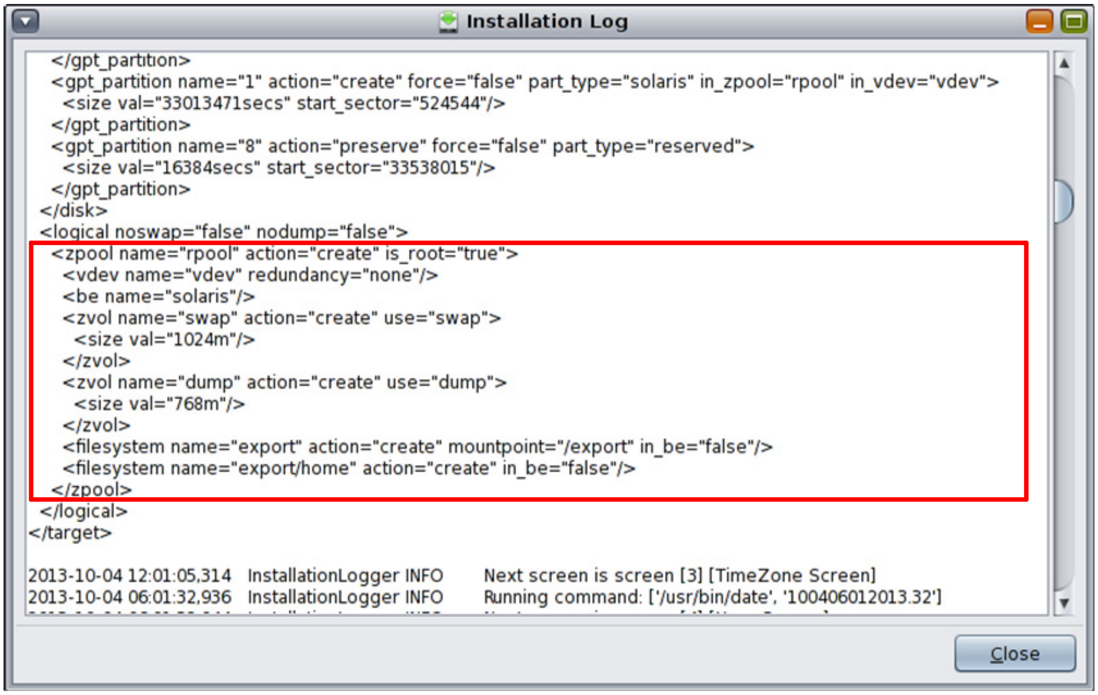
Reviewing the Installation Log



An example of the installation log is shown in the 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.

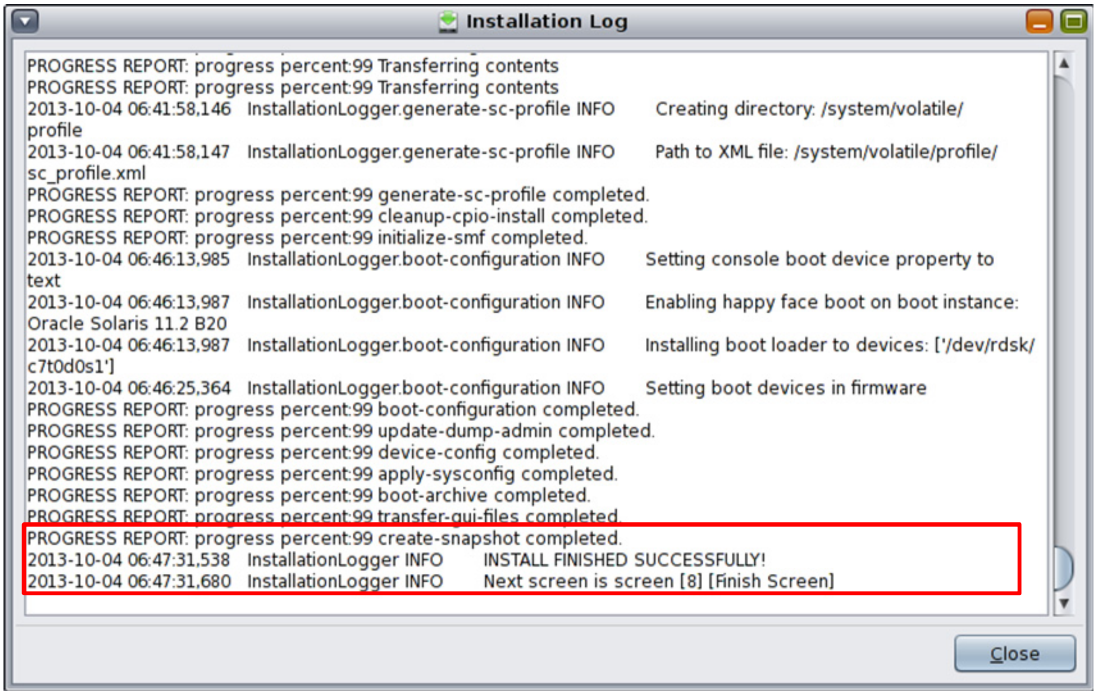
During the first part of the installation process, the configuration settings that you provided during the installation process are captured and applied to the target device.

Reviewing the Installation Log



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

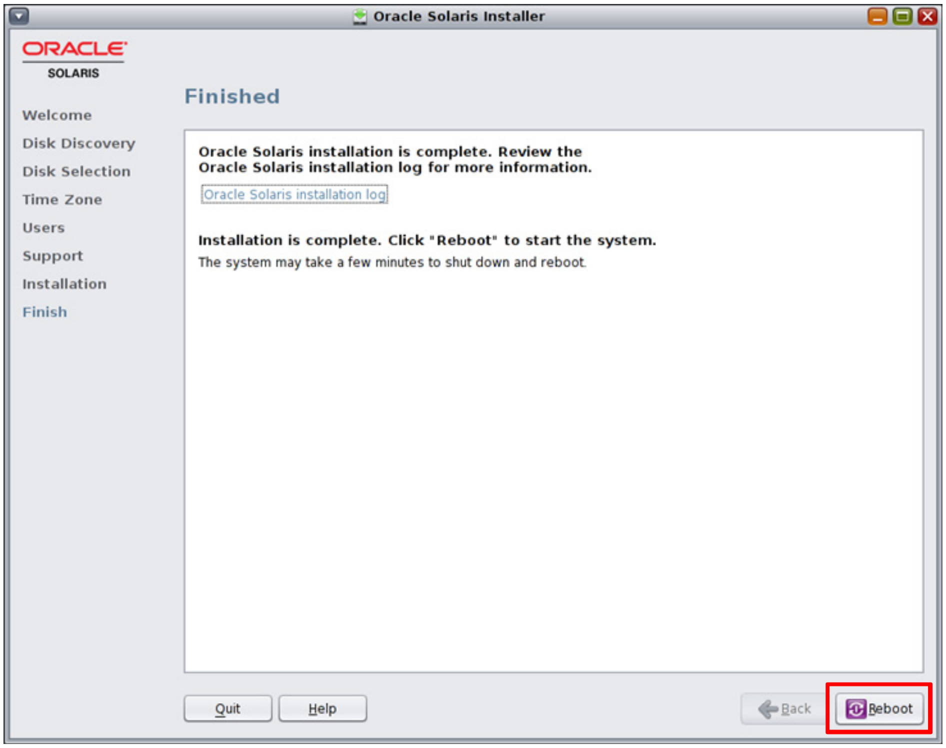
Reviewing the Installation Log



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 that 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.

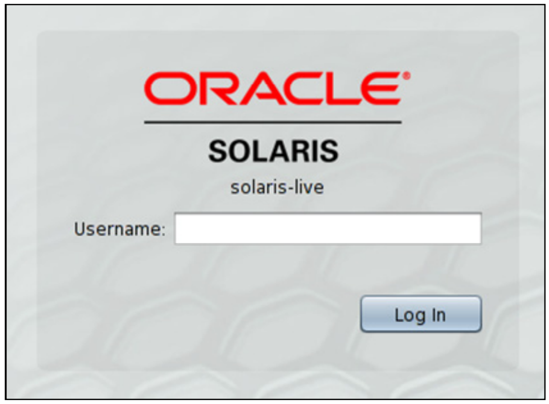
Rebooting the System



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 later in this lesson.

Now you perform your own Live Media installation in Practice 2-1.

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

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 that are designed to reinforce the concepts introduced in the lecture portion of this lesson. You have the chance to perform the following tasks:

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

You find Practice 2-1 in your *Activity Guide*. It should take you about 25 minutes to complete.

Lesson Agenda

- Introduction to Oracle Solaris 11 OS
- Planning for an Oracle Solaris 11 OS Installation
- Installing Oracle Solaris 11 OS by Using the Live Media Installer
- **Installing Oracle Solaris 11 OS by Using the Text Installer**
- Verifying the OS Installation

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 with, you are prompted to select the keyboard layout and language (just as you did for the Live Media GUI installation), after which you see the text installer menu as shown in the 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 for displaying the text installer correctly in case the default terminal type does not work
- **Reboot:** Enables you to reboot the system after the installation has completed

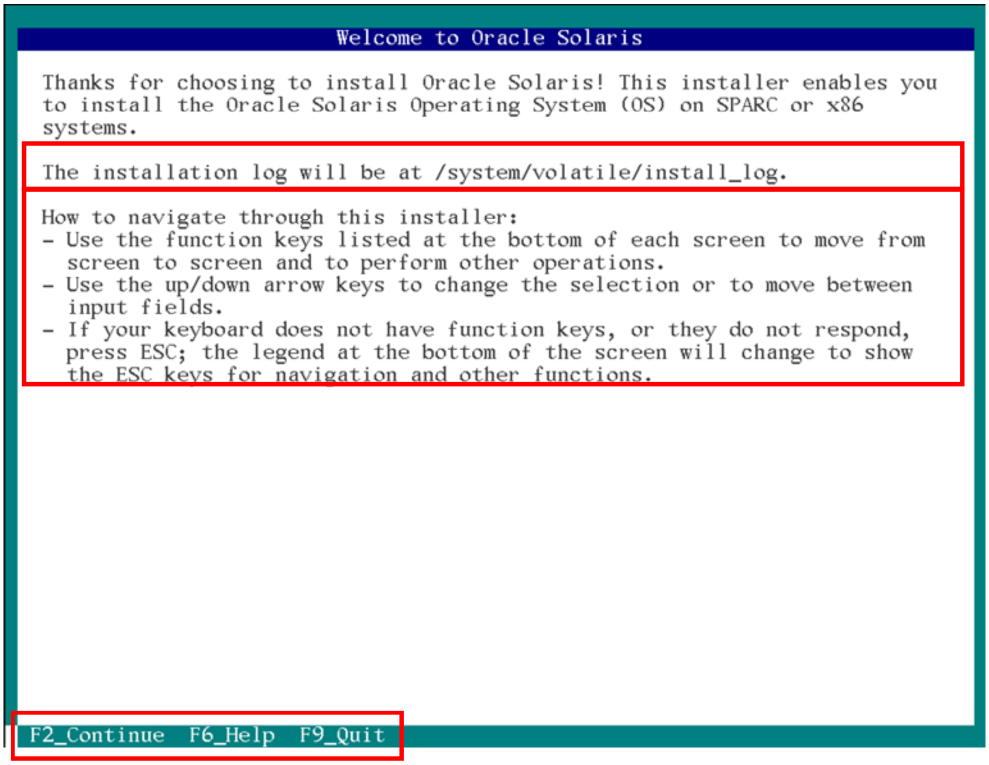
Initiating Installation with the Text Installer

Select option 1 to initiate 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 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



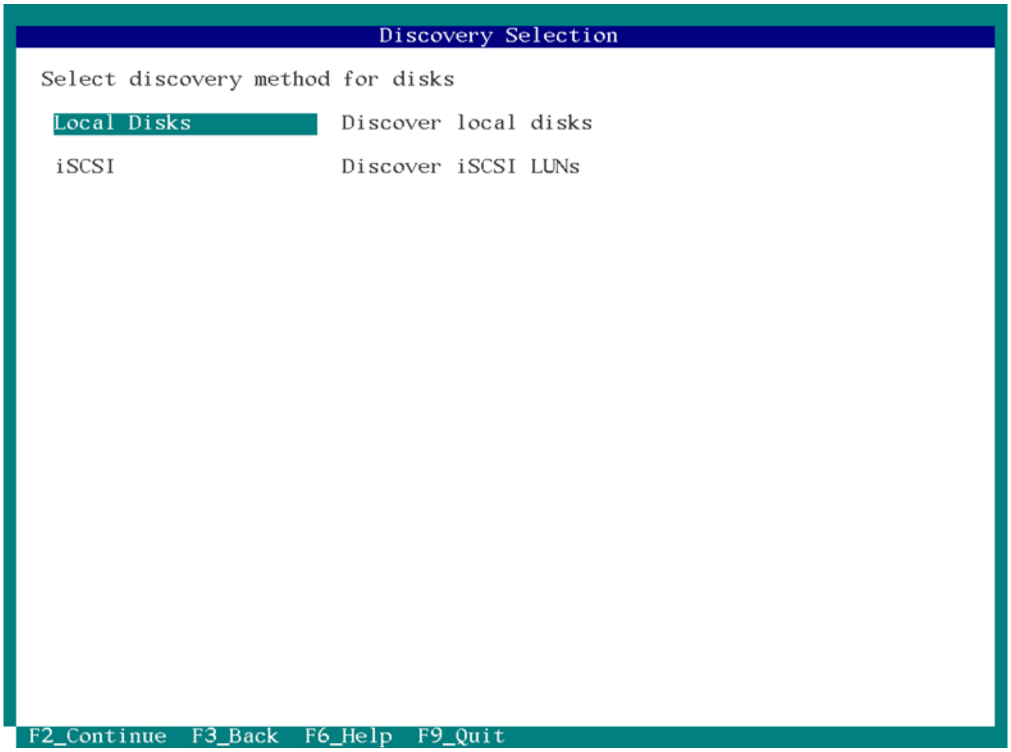
This screen provides you with the temporary location of the installation log (`/system/volatile/install_log`), as well as instructions on how to navigate through the installer by using the function keys that are located at the bottom of the screen and the up and down arrow keys.

You begin the installation by providing configuration data for the disk, time zone, locale, and users. The information that 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, your task is finished.

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 the Discovery Method

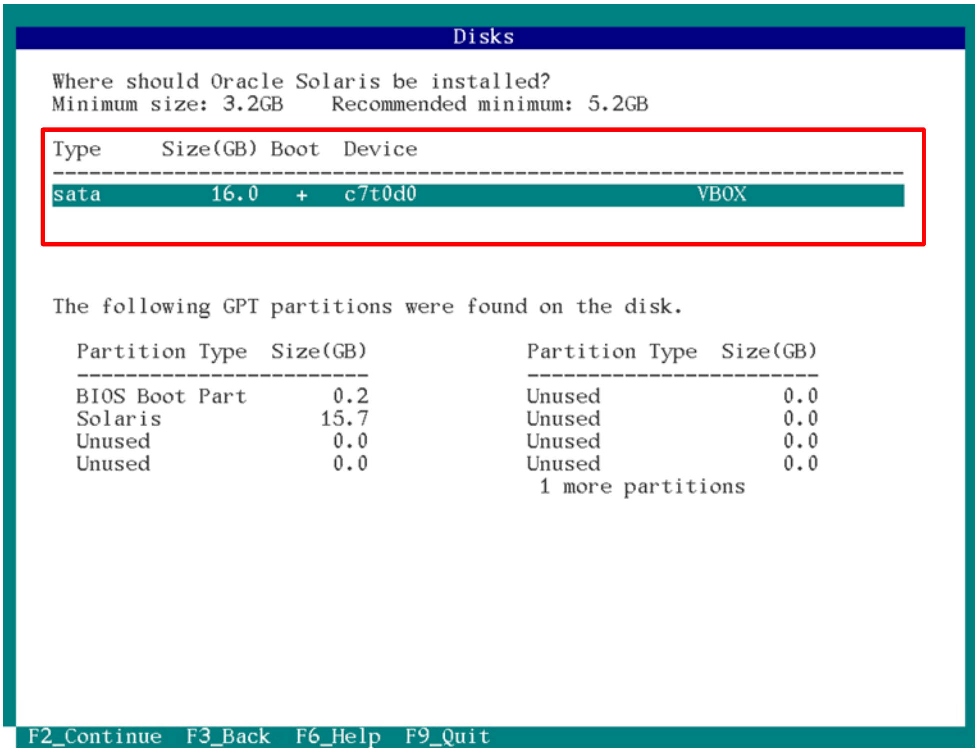


Starting with the Oracle Solaris 11.1 release, the ability to install to iSCSI target LUNs has been included in the text installer. Administrators can choose between installing on local disks or iSCSI disks.

- **Local Disks:** Refers to disks that are attached to the computer, including internal and external hard disks. The "Local Disks" option is selected by default.
- **iSCSI:** Refers to remote iSCSI disks that you connect to by using DHCP auto-discovery or by manually specifying a target IP address, iSCSI target name and LUN, and initiator name. If you want the installer to search for remote disks that are accessible over a network by using the iSCSI standard, you select the "iSCSI" option.

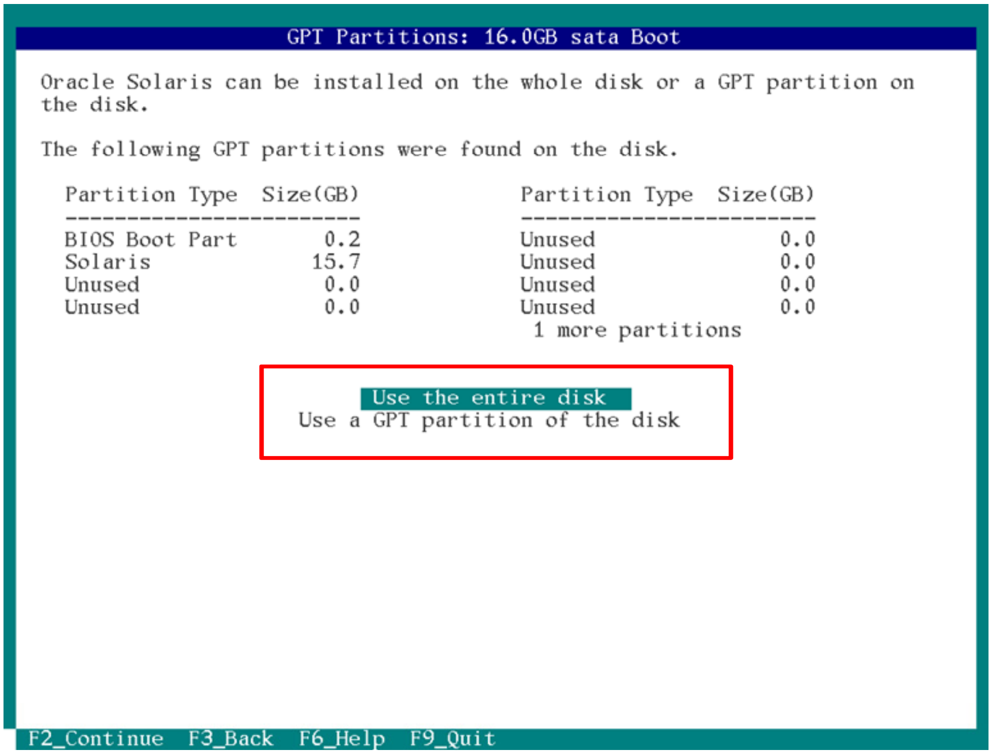
For more information, refer to
http://docs.oracle.com/cd/E36784_01/html/E36800/guitask.html#scrolltoc.

Selecting a Disk



From the Discovery Selection screen, you are taken to the Disks screen, where you are prompted to select the location of the Oracle Solaris OS installation. To select the highlighted disk and continue to the next screen, press F2.

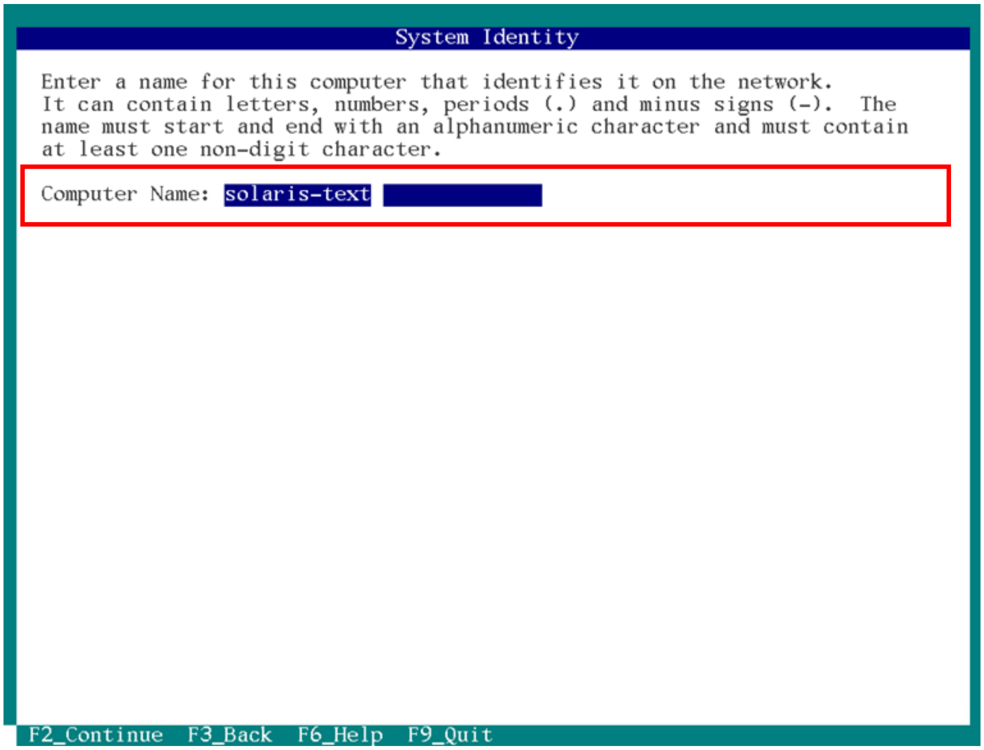
Selecting an Fdisk Partition



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

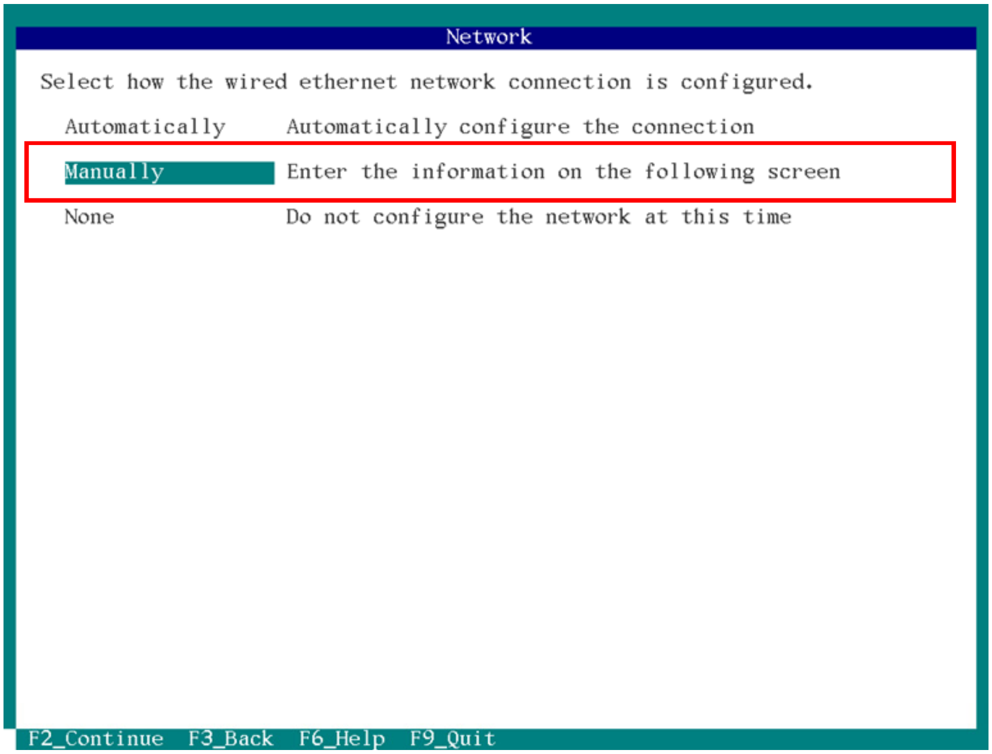
If you choose to modify the partition layout, refer to the guidelines for partitioning the system at http://docs.oracle.com/cd/E36784_01/html/E36800/guidepart.html#scrolltoc.

Providing a System Identity



On the System Identity screen, you provide a computer name to identify the system on the network. When you have typed the desired computer name, press F2 to continue.

Selecting a Network

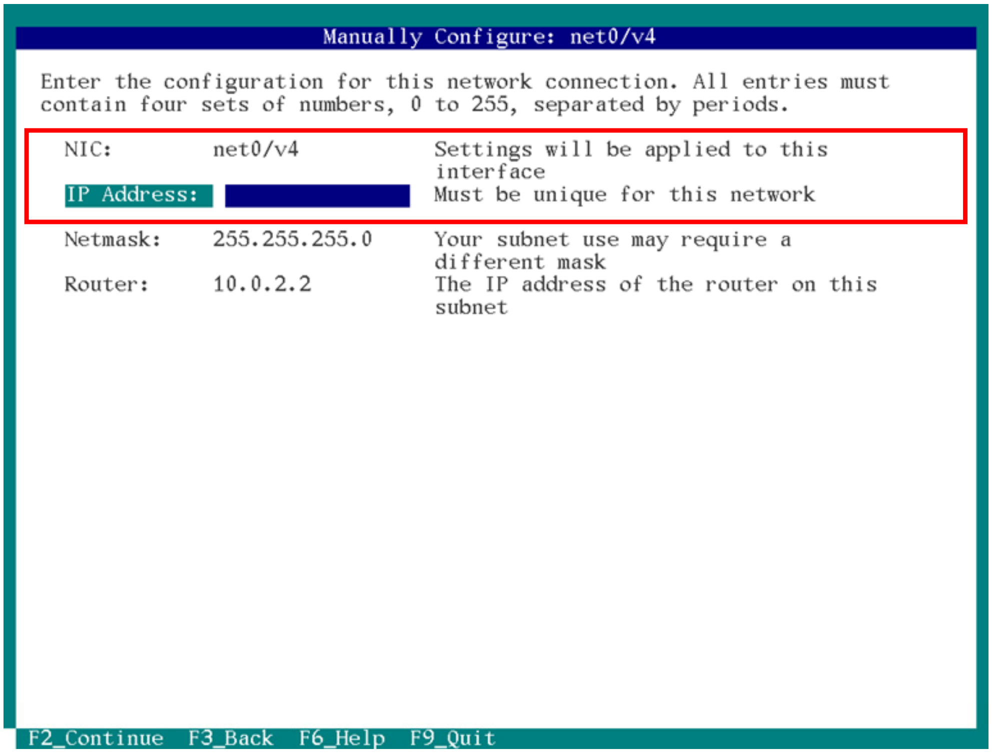


The next screen is the Network screen, where you select the wired Ethernet network connection configuration (Automatically, Manually, or None).

- The Automatically option automatically configures the network connection for you. If you want to have the network configured automatically, Oracle Solaris configures it by using a feature called reactive networking 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 learn more about network configurations, both physical and virtual, in the lesson titled "Administering the Network."
- The Manually option enables you to manually configure the network connection by responding to the prompts presented on 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, use the down arrow key to move 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.

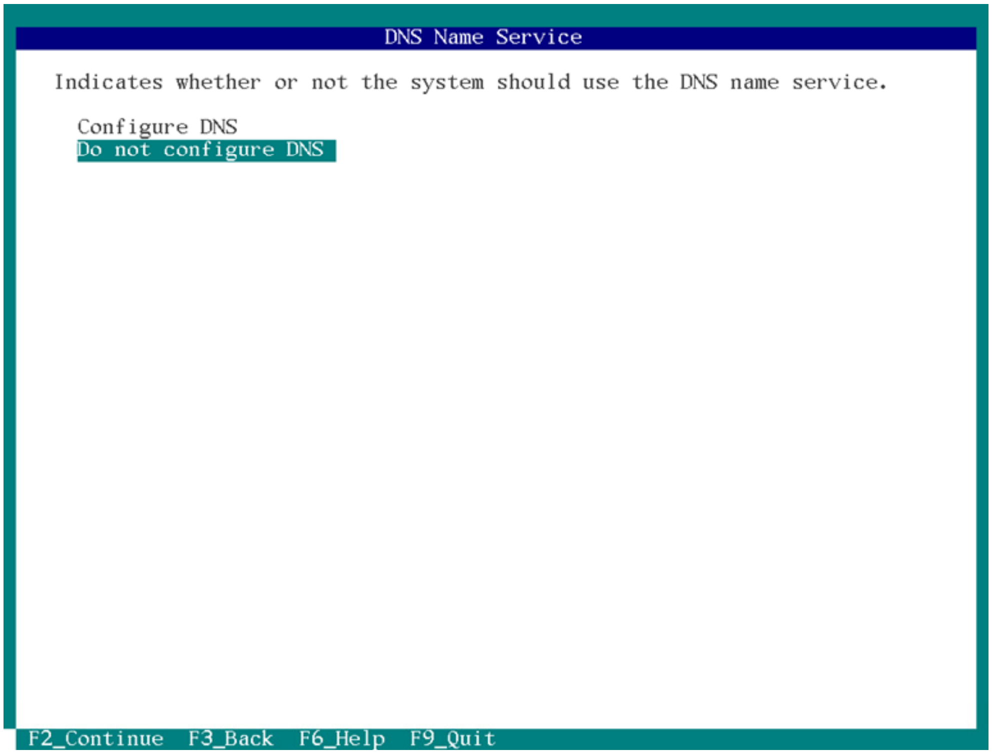
Manually Configuring the Network



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.

Note: IP address and netmask are required fields. Router is an optional field.

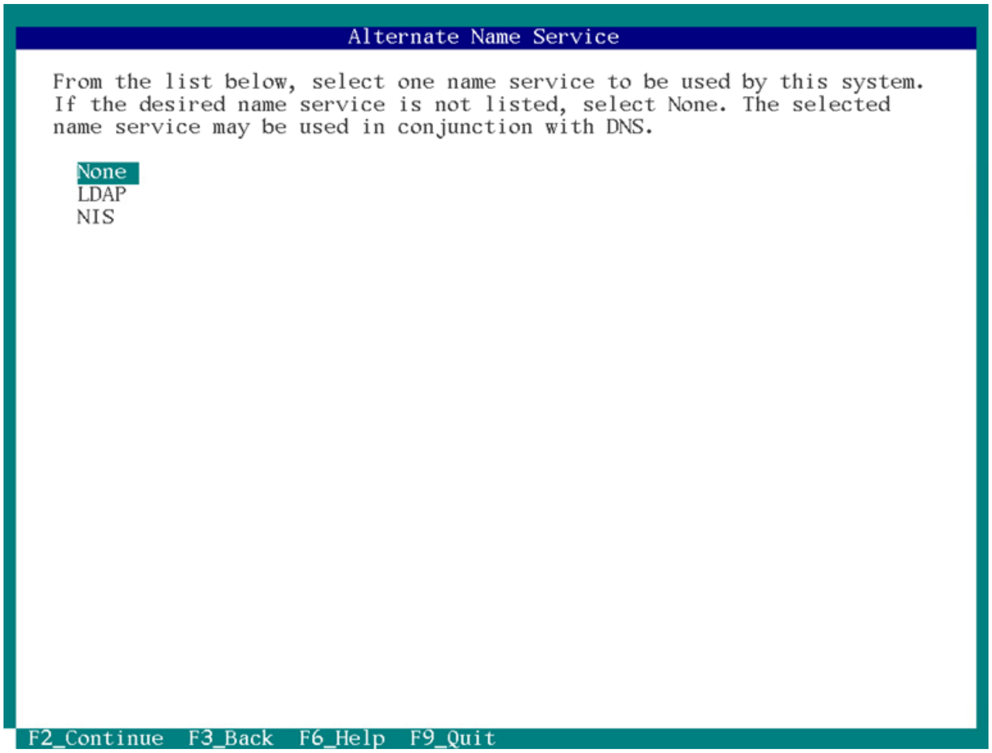
DNS Name Service



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 do not configure DNS.

Note: If you choose to configure a DNS name server, you need to provide at least one IP address for the DNS server or servers to be used by the system and at least one domain name to be searched when a DNS query is made.

Alternate Name Service

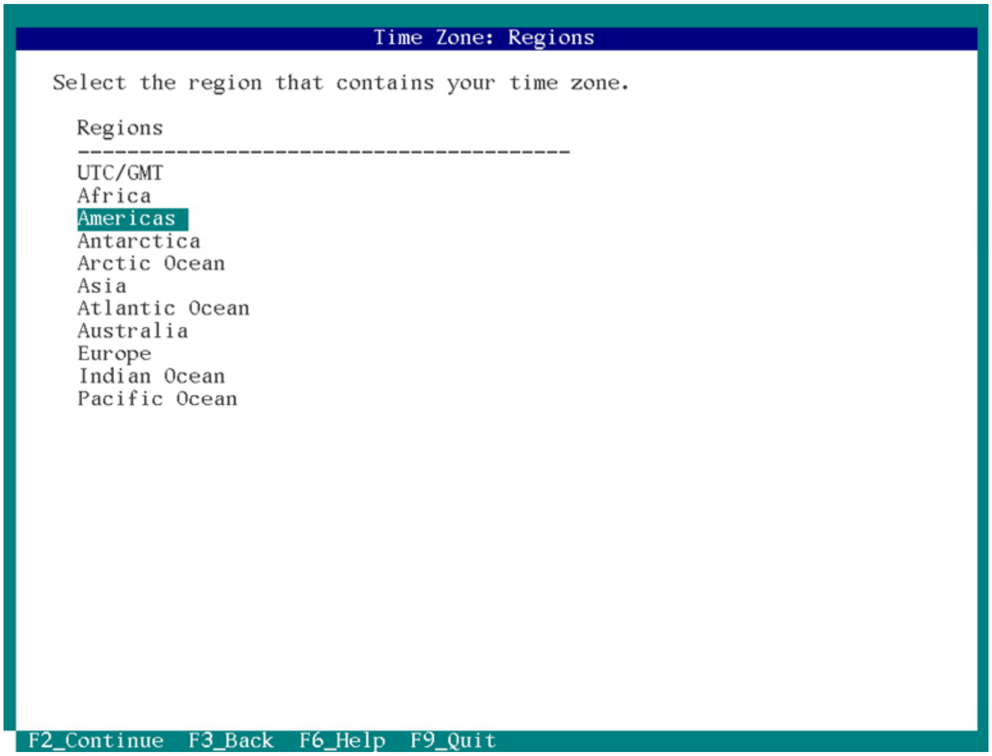


The next screen enables you to select an alternate name service, such as LDAP or NIS. For this example, you select None.

Notes

- If you chose to configure DNS in the previous step, LDAP or NIS would be set up as alternate name services in addition to DNS. For more information about how to set up LDAP or NIS, refer to http://docs.oracle.com/cd/E36784_01/html/E36800/texttask.html#scrolltoc.
- If you choose to configure LDAP on the system without an LDAP profile, select None instead of selecting LDAP. Then, configure LDAP manually after the installation is complete.
- If no network naming services are selected, network names are resolved by using standard name source files such as `/etc/hosts`(4). For further information, refer to the `nsswitch.conf`(4) man page.

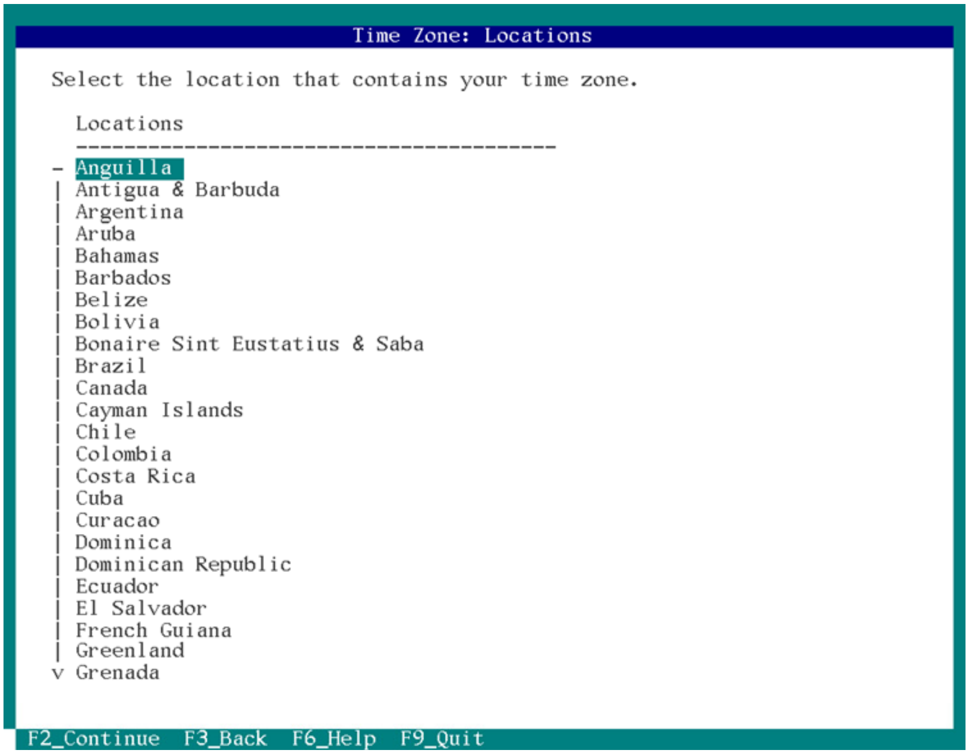
Selecting Time Zone: Regions



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

In this example, you select Americas.

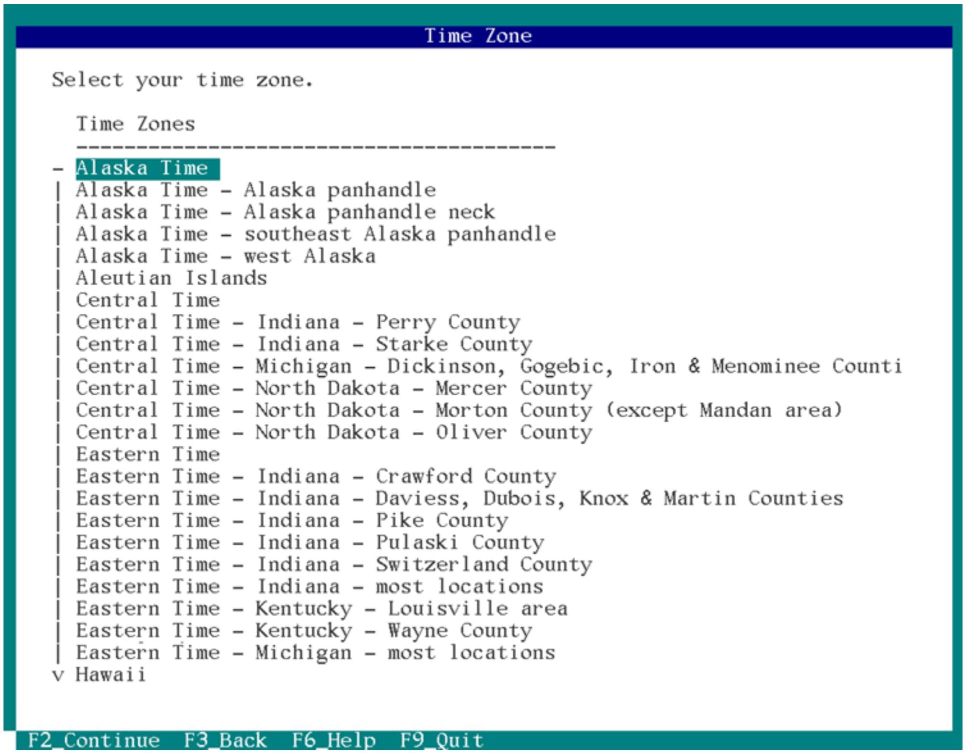
Setting Time Zone: Locations



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

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

Selecting the Time Zone

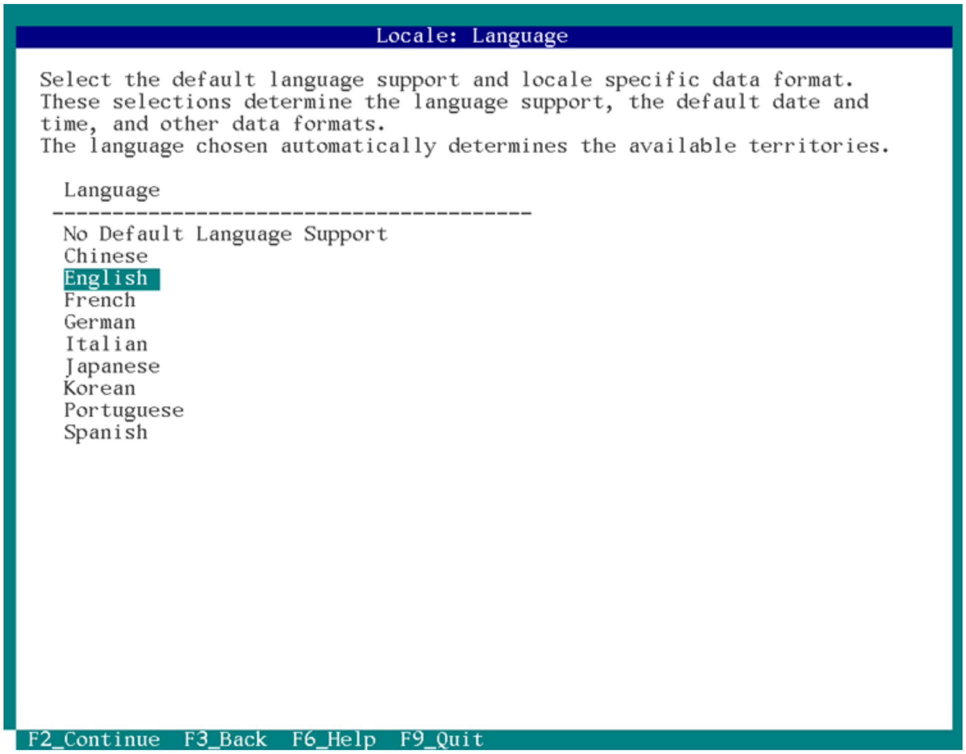


ORACLE

Next, you 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, use the down arrow key until your time zone is highlighted, and then press F2.

To support the example installation, you select Mountain Time (not shown in the slide).

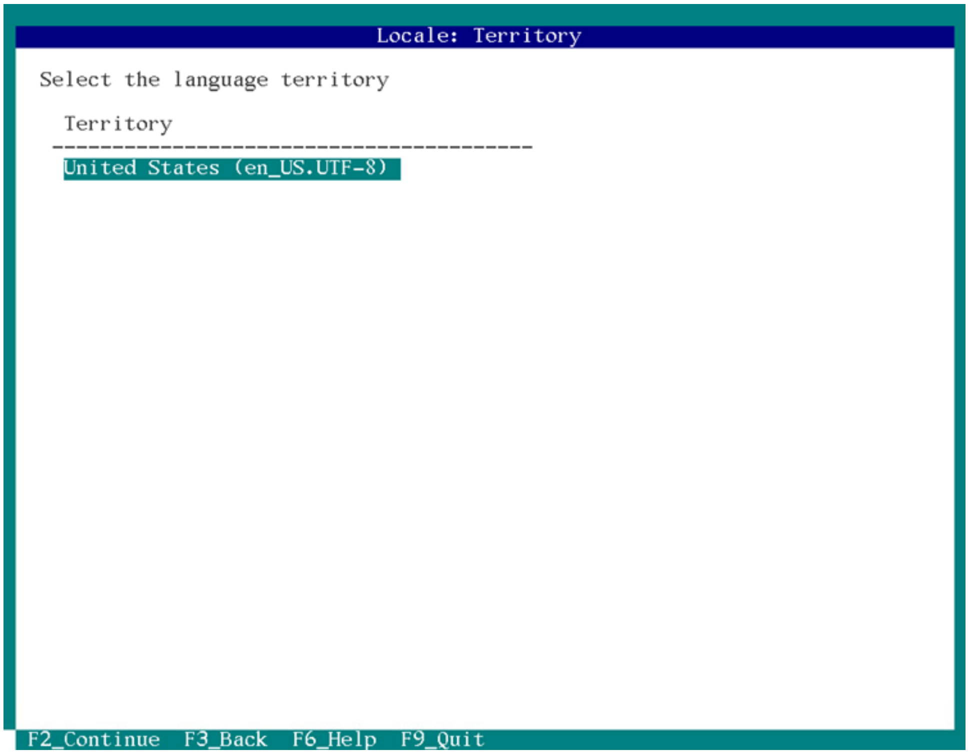
Selecting the Language



Next, you see the Locale: Language screen where you select the default language support. Based on the language that you select on this screen, the available territories are automatically listed on the next screen.

To select the appropriate language, use the down arrow key until your desired language is highlighted, and then press F2.

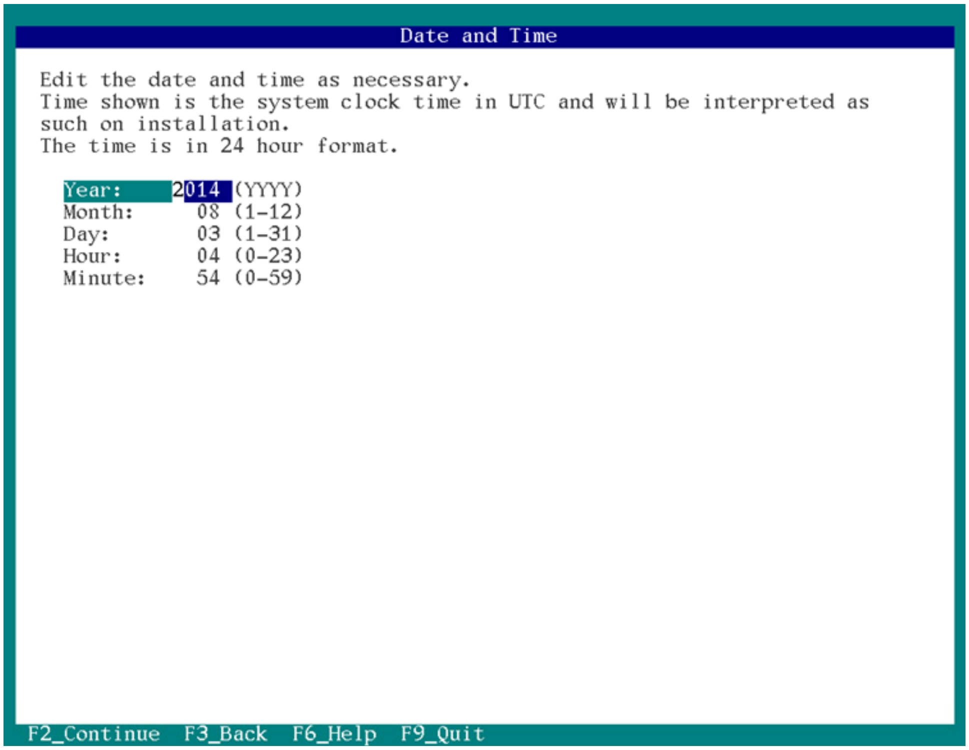
Selecting the Territory



Next, you see the Locale: Territory screen where you select the language territory. Based on the language that you selected on the previous screen, the available territories are automatically listed on this screen.

To select the appropriate territory, use the down arrow key until your desired territory is highlighted, and then press F2.

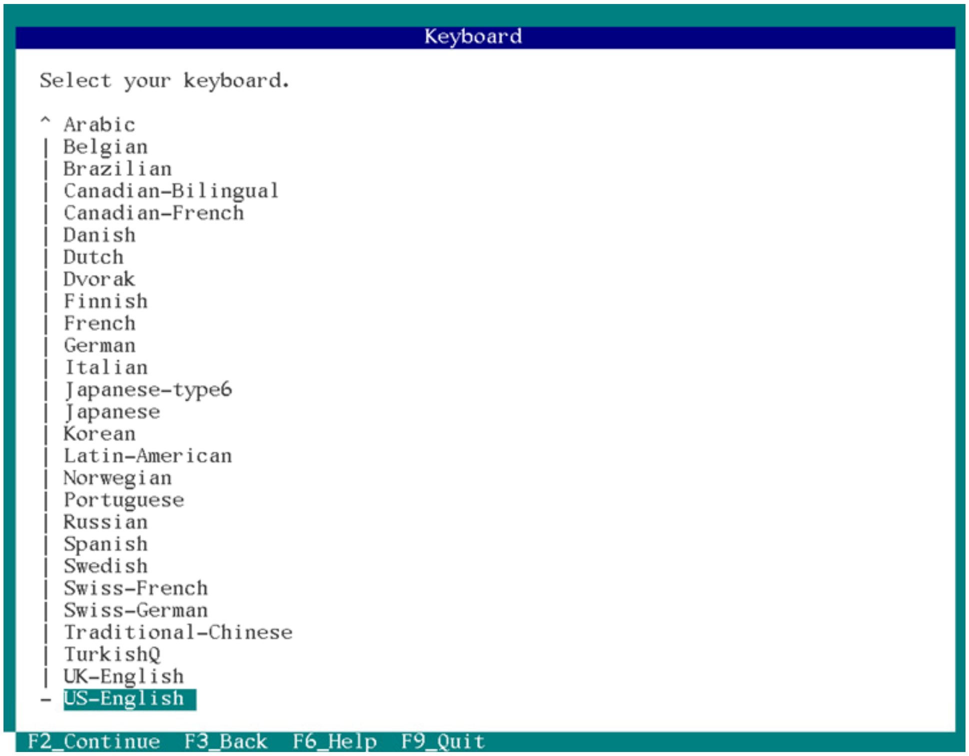
Setting the Date and Time



ORACLE

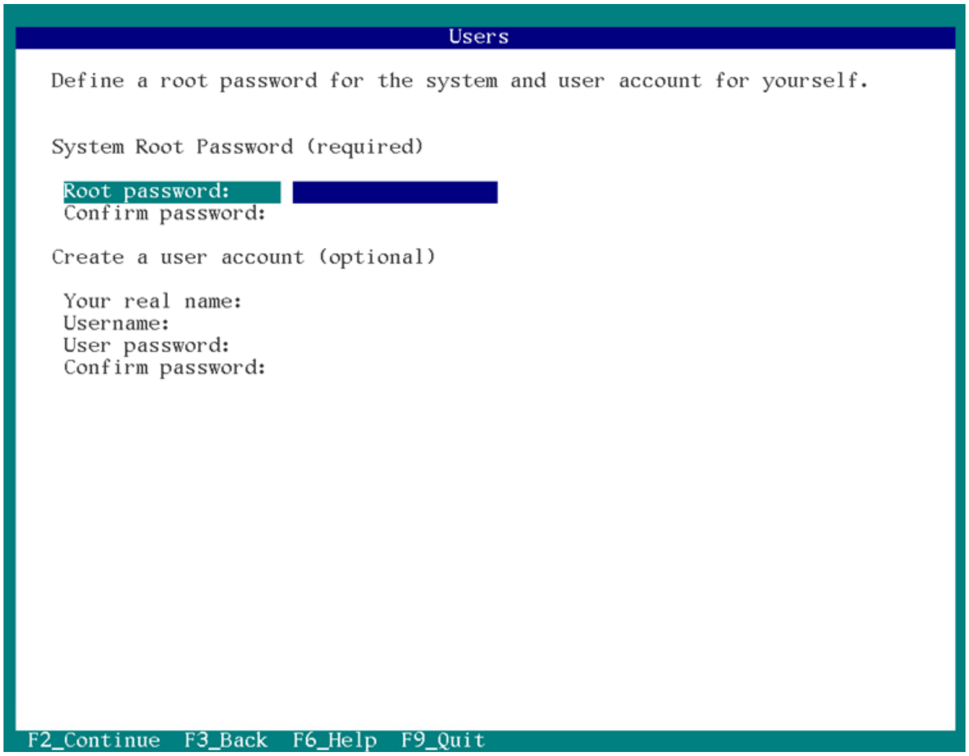
The Date and Time screen appears next. Review the information that is presented and edit it as required. Note that the time is in the 24-hour format. After you have made the necessary edits, press F2 to continue.

Selecting the Keyboard



Next you see the Keyboard screen. To select the desired keyboard layout, use the down arrow until it is highlighted, and then press F2.

Providing User Information

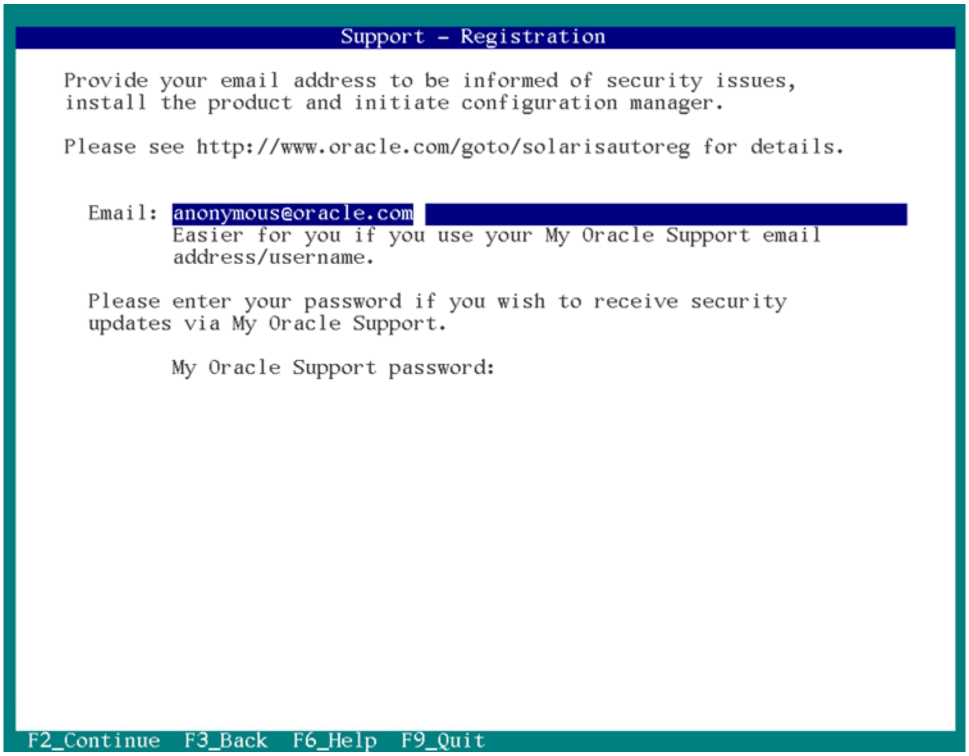


ORACLE

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

Note: Providing a root password is mandatory. However, creating a user account is optional. 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.

Registering to My Oracle Support



On the Support Registration screen, you can provide your email address and password if you want to receive security updates from My Oracle Support. Press F2 to save and press F2 again to proceed to the next screen.

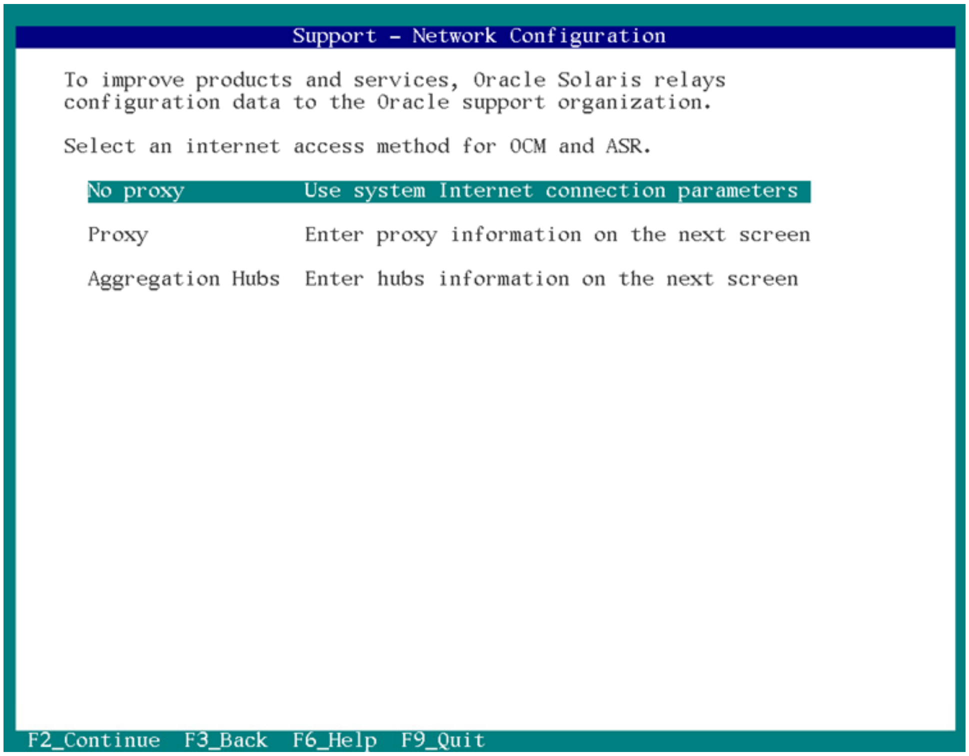
The default Support Registration screen provides an anonymous registration address. If you use this anonymous address with no password, My Oracle Support (MOS) will receive information about the installed system's configuration, but will not receive any of your information when the system configuration is uploaded to the Oracle Support organization.

Alternatively, you can register for security updates or disconnect OCM as follows:

- Replace the anonymous email address with your My Oracle Support login ID and password. Use this option if you want to see your information in My Oracle Support and receive security updates. With this option, ASR will also be started.
- Delete the anonymous email address and leave the field blank. OCM will start in a disconnected mode. No data will be sent to My Oracle Support. Or, if you delete the anonymous email address and replace it with another email address other than your MOS login ID, OCM will send data to Oracle Support in an unauthenticated mode.

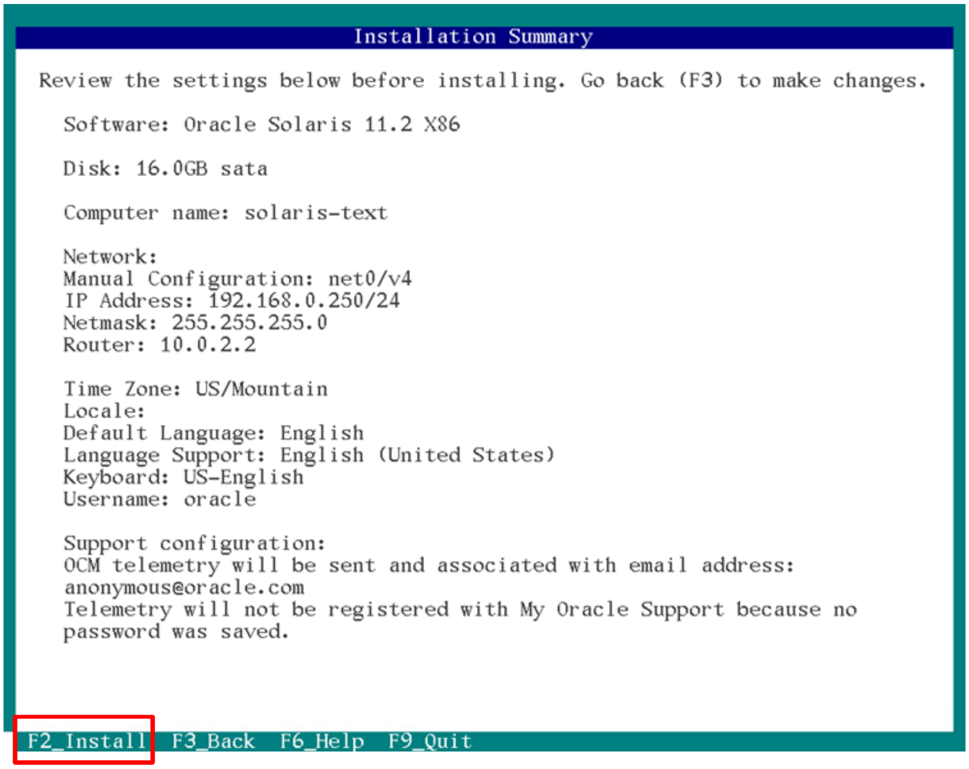
For further information, refer to http://docs.oracle.com/cd/E36784_01/html/E36800/ocm.html#scrolltoc.

Support Network Configuration



"No proxy" is selected by default. Press F2 to accept the default option and proceed to the next screen.

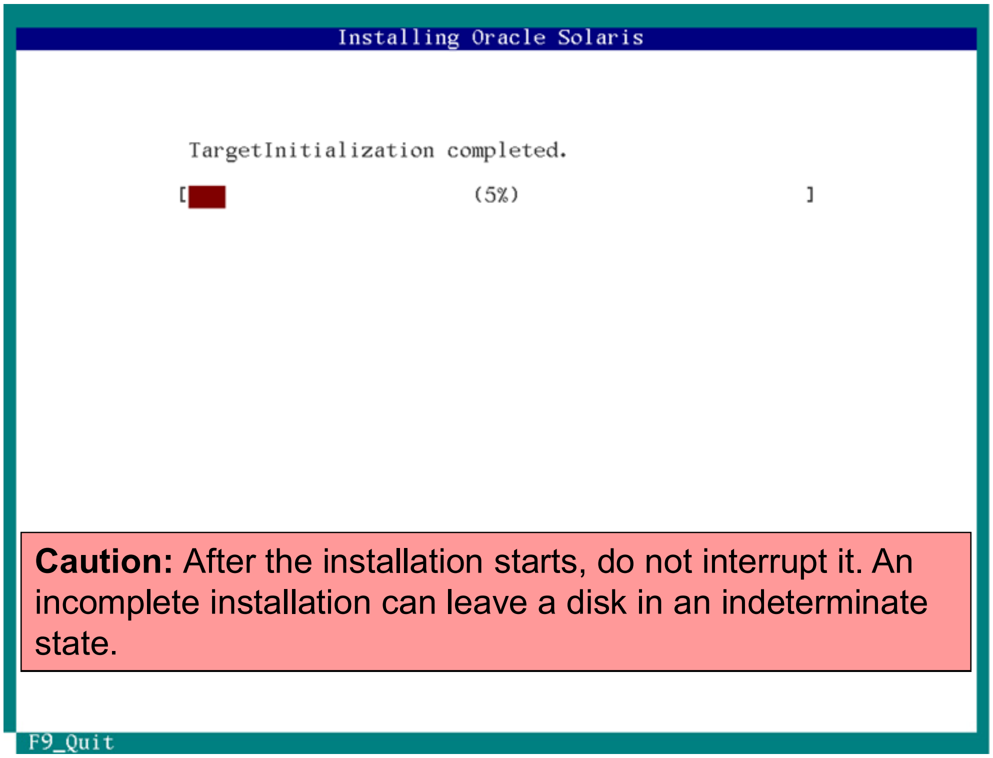
Reviewing the Installation Summary



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

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

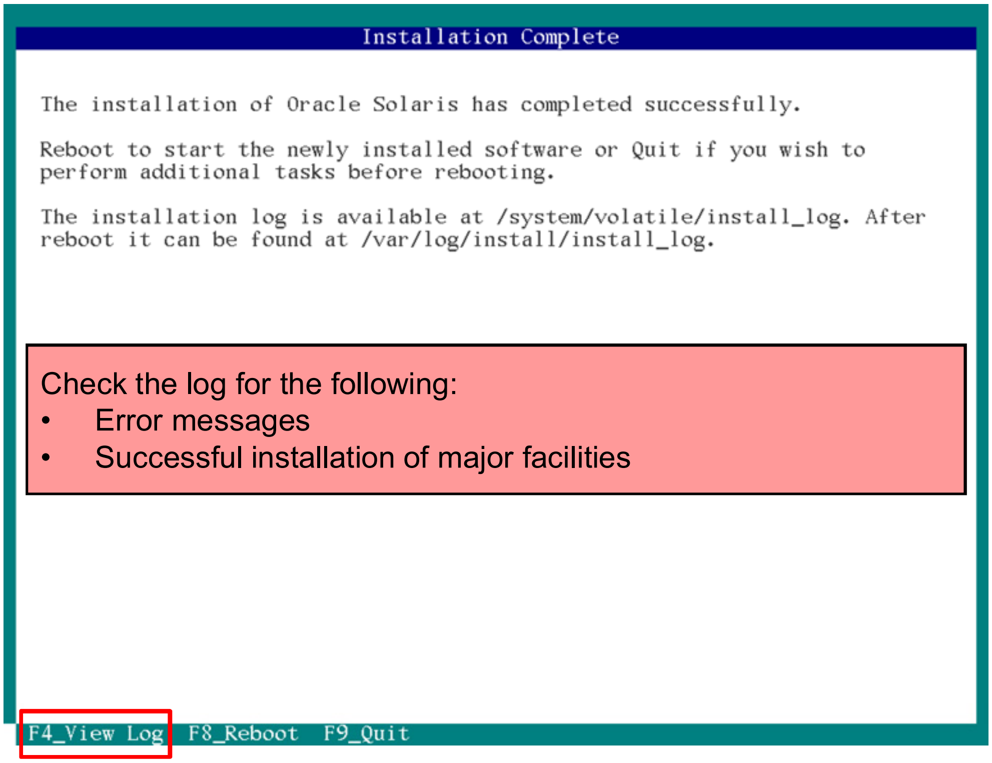
Monitoring the Installation



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.

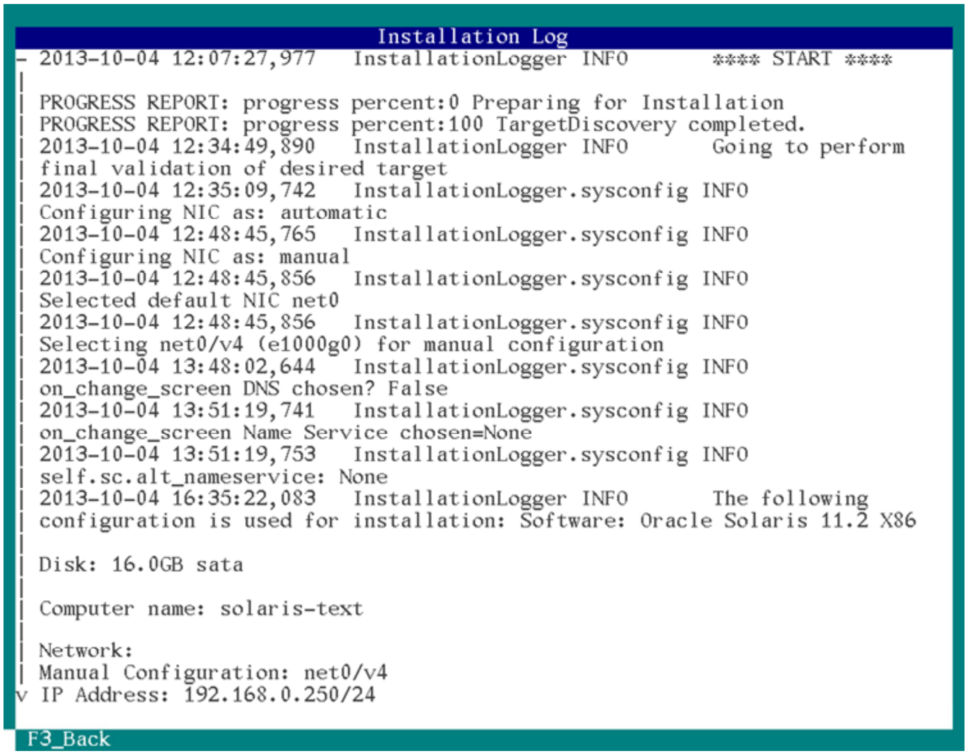
Verifying the Installation



When the installation concludes, the Installation Complete screen appears. This screen provides you with access to the installation log (similar to the one that you observed during the Live Media installation), which gives you 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/log/install/install_log` after reboot.

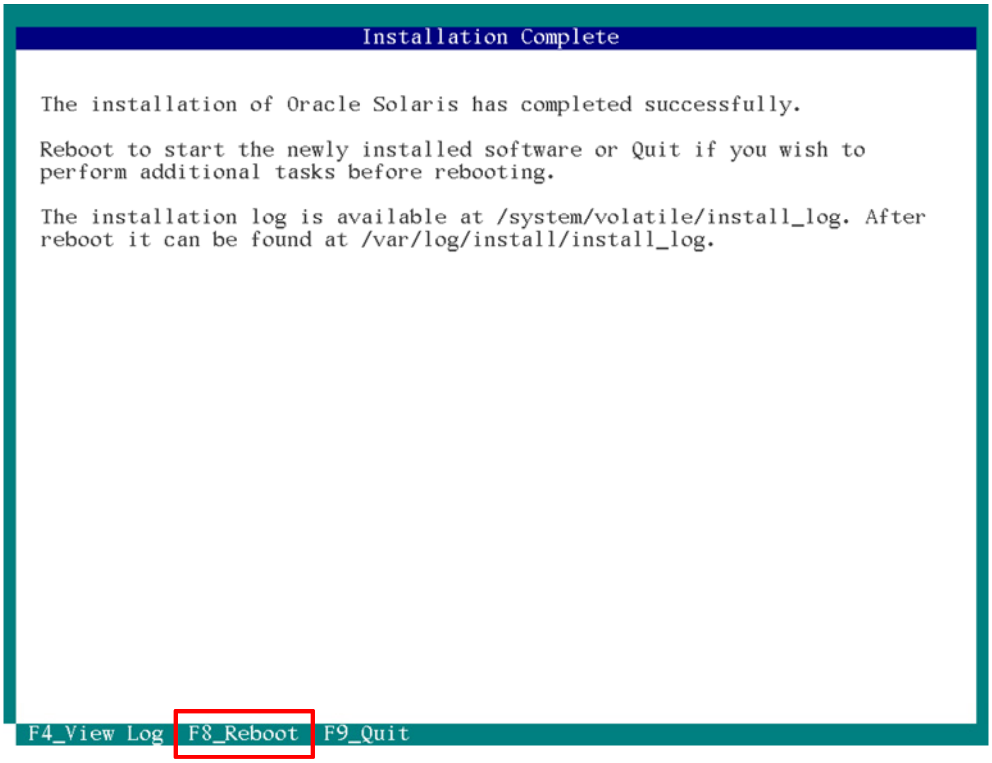
Reviewing the Installation Log



An example of the installation log for the text installation is shown in the slide. Similar to the Live Media installation log, the text installation 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 that 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 11.2 64-bit
Copyright (c) 1983, 2014, Oracle and/or its affiliates. All rights reserved.
Loading smf(5) service descriptions: 202/202
Configuring devices.
Hostname: solaris-text

solaris-text console login: _
```


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 text installation in Practice 2-2.

Practice 2-2 Overview: Installing Oracle Solaris 11 by 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 25 minutes to complete.

Lesson Agenda

- Introduction to Oracle Solaris 11 OS
- Planning for an Oracle Solaris 11 OS Installation
- Installing Oracle Solaris 11 OS by Using the Live Media Installer
- Installing Oracle Solaris 11 OS by Using the Text Installer
- **Verifying the OS Installation**

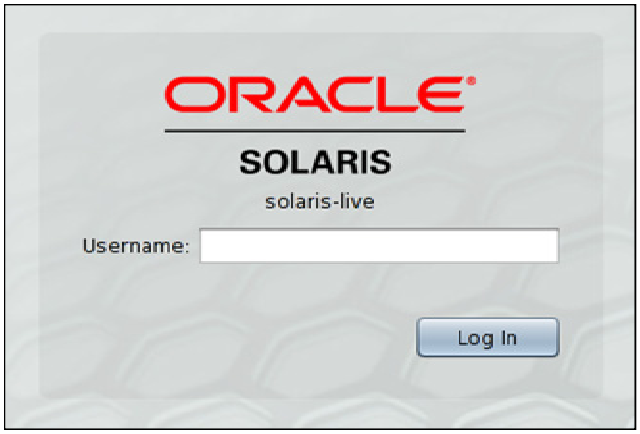
Verifying the Operating System Installation

- Verifying login information
- Using first time login assistant
- Verifying the system’s host name and host ID
- Displaying basic system information
- Displaying a system’s release information
- Displaying disk configuration information
- Displaying the installed memory size
- Displaying the disk space information
- 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 OS 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.

Verifying the Login Username

Live Media



Text Installation

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


Live Media GUI Environment (`solaris-live` system)

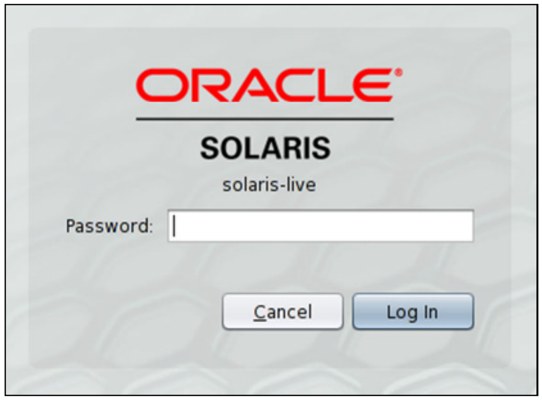
On the login screen (as shown at the top of the slide), enter the username that you provided during installation, and then click Log In. If the username is correct, the password screen appears.

Text Installation Environment (`solaris-text` system)

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

Verifying the Login Password

Live Media



Text Installation

```
Password:
Oracle Corporation  SunOS 5.11  11.2  June 2014
oracle@solaris-text:~$
```

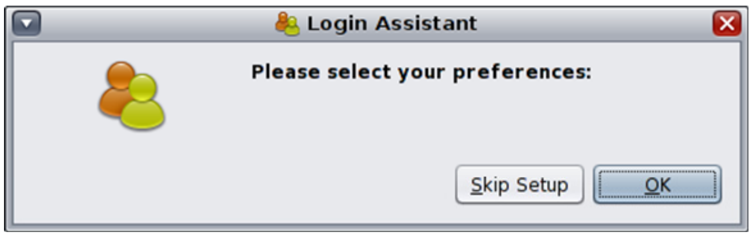

Live Media GUI Environment (`solaris-live` system)

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

Text Installation Environment (`solaris-text` system)

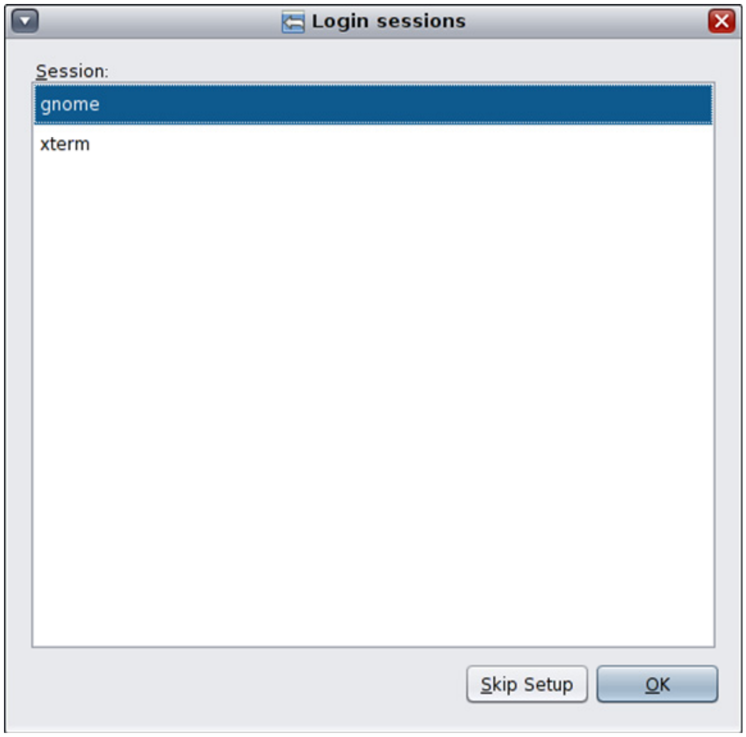
To check the login password, go to the password prompt, enter the password that you provided during installation, and press Enter. If the password is correct, you are presented with the operating system release information and the command-line user prompt (as shown in the code example in the slide).

Live Media GUI: Using the First Time Login Assistant



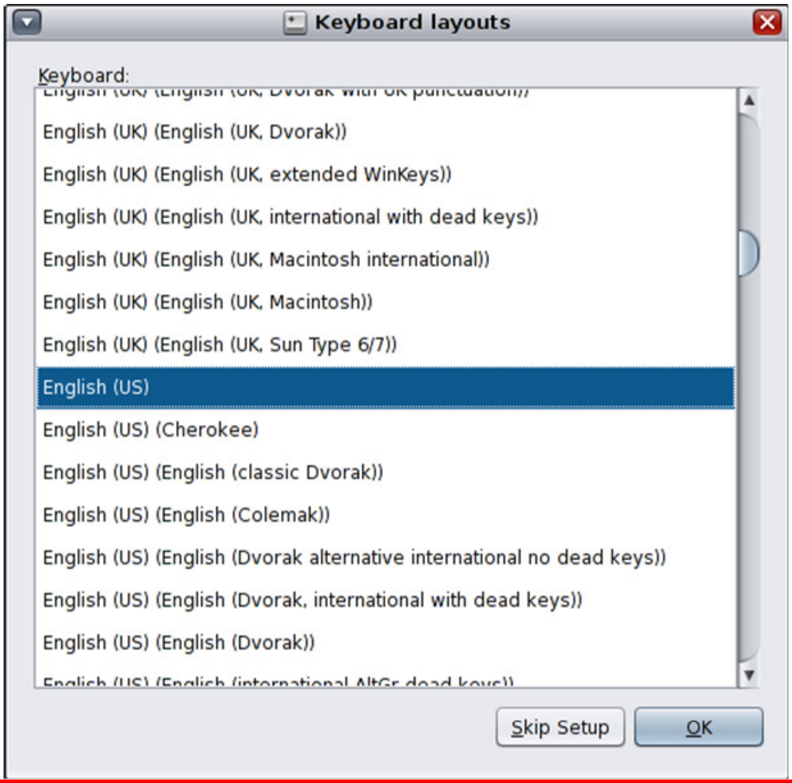
If you are in the Live Media GUI environment, you are presented with the First Time Login Assistant the first time that you log in. The First Time Login Assistant enables you to select a login session type and language. If you think you do not need this type of assistance, you can click Skip Setup. If you click OK, the Login Session window appears.

Live Media GUI: Selecting a Login Session



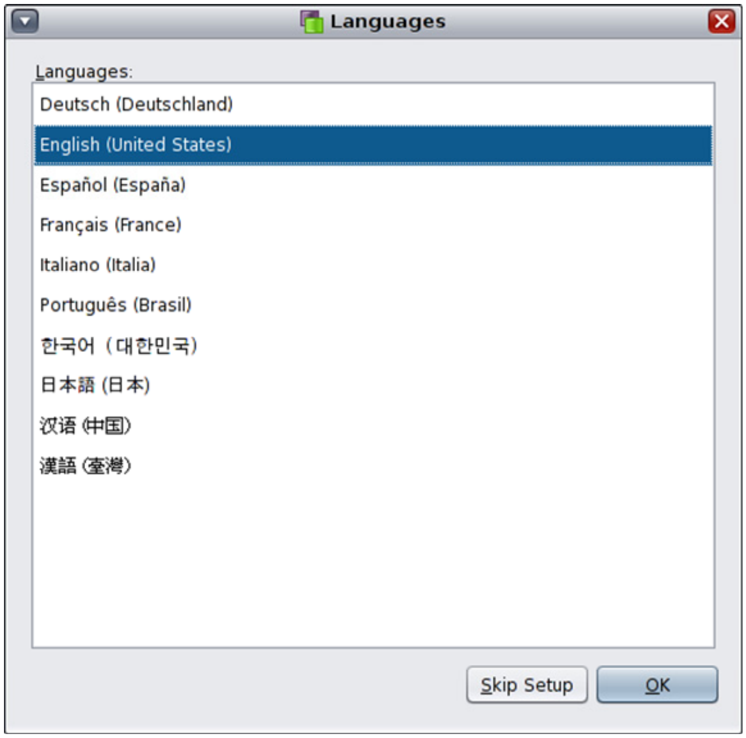
The “Login sessions” window allows you to select the type of login session that you want to use. Your choices are gnome or xterm. Gnome is a desktop session and xterm is a terminal window. To select a session, highlight it and click OK. You select gnome.

Live Media GUI: Selecting a Keyboard Layout



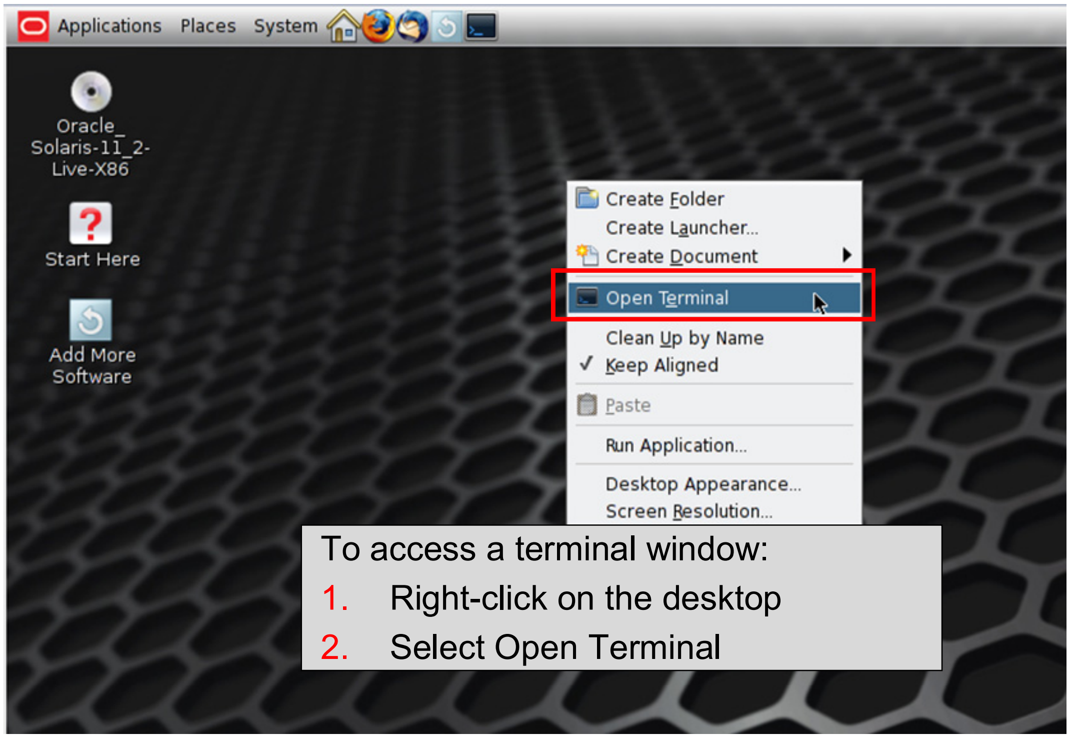
The next window is the Keyboard Layouts window. To select a keyboard, highlight it and click OK.

Live Media GUI: Selecting a Language



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

Live Media GUI: Accessing a Terminal Window from Gnome



After 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 context menu, and then select Open Terminal.

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

Verifying the Host Name and Host ID

To display the host name, use the `hostname` command.

```
$ hostname
solaris-live
```

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

To display the host ID, use the `hostid` command.

```
$ hostid
00809442
```


When you are in the terminal window, the first thing that 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 installation machine, your host name would appear as `solaris-text` (if that is the name you gave the system during installation).

The `hostid` command prints the identifier of the current host in hexadecimal. This numeric value is likely to differ when `hostid` is run on a different machine.

Displaying Basic System Information

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

```
$ uname -a
SunOS solaris-live 5.11 11.2 i86pc i386 i86pc
```

This system's basic information is as follows:

- Operating system: `SunOS`
- Hostname: `solaris-live`
- Release: `5.11`
- Version: `11.2`
- Node name: `i86pc`
- Hardware name: `i386`
- Processor type: `i86pc`

When you have verified the host name, the next thing that 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 11 or Oracle Solaris 11.1. Capturing this information about your system for baseline purposes is extremely important for updating software packages, which is discussed in detail in the follow-on course, *Oracle Solaris 11 Advanced System Administration*.

Note: You can also run the `uname` command with specific options to display any one of the information items individually. Refer to 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 11.2 X86
   Copyright (c) 1983, 2014, Oracle and/or its affiliates.
                      All rights reserved.
                    Assembled 23 June 2014
```


The Oracle Solaris 11 operating system has a `/etc/release` file, which contains the following information about the system:

- Complete operating system name
- Version of the release
- Hardware architecture
- Copyright
- Date on which the release was assembled

You use the `cat /etc/release` command to display the contents of this file.

Note: Use the `cat /etc/release` command (instead of `uname -a`) to get current update release information.

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. c1t0d0 <ATA- VBOX HARDDISK-1.0-16.00GB>
          /pci@0,0/pci8086,2829@d/disk@0,0
Specify disk (enter its number): 0
selecting c1t0d0
 [disk formatted]
/dev/dsk/c1t0d0s1 is part of active ZFS pool rpool. Please see
zpool(1M).

<continued on next page>
```

Note: The `format` utility requires `root` role privileges.

You use the `format` command to display information about the disks that are on the system. Recall that you used the entire disk for installation and that the disk name was `c1t0d0`.

The `format` command 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` role, you can use the `su -` command.

Note: `su` stands for “switch user.”

In the slide example, only one disk has been configured in the system: `c1t0d0`. To see how a disk is formatted, you provide the disk number (in this example, `0`) for the `Specify disk (enter its number)` prompt, and then press the Enter key.

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

```
Volume name = <        >
ascii name  = <ATA-VBOX HARDDISK-1.0-16.00GB>
bytes/sector = 512
sectors =33554431
accessible sectors = 33554398

Part         Tag    Flag   FIRST Sector      Size      Last Sector
  0   BIOS boot      wm             256    256.00MB      524543
  1         usr      wm          524544    15.74GB     33538014
  2  unassigned      wm               0       0             0
  3  unassigned      wm               0       0             0
  4  unassigned      wm               0       0             0
  5  unassigned      wm               0       0             0
  6  unassigned      wm               0       0             0
  8   reserved       wm         3353801      8.0MB       33554398
format> quit
#
```

ORACLE

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 you selected to install the operating system) has nine partitions, with the first partition (partition `0`) being allocated to contain the root file system. Partitions `2`, `3`, `4`, `5`, and `6` are currently unassigned. Partition `1` represents the whole disk. Partition `8` contains the boot program and is used in the booting process.

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.

For information about how to partition the system, refer to the following URLs:

- http://docs.oracle.com/cd/E36784_01/html/E36834/disksconcepts-29477.html#scrolltoc
- http://docs.oracle.com/cd/E36784_01/html/E36834/disksprep-31030.html#scrolltoc

Displaying Installed Memory Size

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

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


Next, you want to capture for your baseline documentation the total amount of physical memory installed on the system. Oracle Solaris provides the `prtconf` command to print the system configuration that was initially created by the lower-level hardware discovery phase. The output of the `prtconf` command usually includes the total amount of memory, and the configuration of system peripherals that are formatted as a device tree. Refer to the `prtconf`(1M) man page for more details.

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, you have 1024 MB of installed memory.

Displaying Disk Space Information

To display space utilization for the file system, use the `df -h` command.

```
# df -h
Filesystem             Size   Used  Available Capacity  Mounted on
rpool/ROOT/solaris      15G   4.0G       9.2G    31%    /
/devices                 0K     0K         0K     0%    /devices
/dev                     0K     0K         0K     0%    /dev
ctfs                     0K     0K         0K     0%    /system/contract
proc                     0K     0K         0K     0%    /proc
mnttab                   0K     0K         0K     0%    /etc/mnttab
swap                   661M   1.4M       659M     1%    /system/volatile
objfs                    0K     0K         0K     0%    /system/object
sharefs                  0K     0K         0K     0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcap1.so.1    13G   4.0G       9.2G    31%    /lib/libc.so.1
fd                       0K     0K         0K     0%    /dev/fd
rpool/ROOT/solaris/var    15G   292M       9.2G     4%    /var
swap                   787M   128M       659M    17%    /tmp
rpool/VARSHARE          15G    48K       9.2G     1%    /var/share
rpool/export            15G    32K       9.2G     1%    /export
rpool/export/home       15G    32K       9.2G     1%    /export/home
rpool/export/home/oracle    15G   871K       9.2G     1%    /export/home/oracle
rpool                   15G   5.0M       9.2G     1%    /rpool
...
(output truncated)
```


Next, you want to capture for your baseline documentation the space utilization on your installed system. Oracle Solaris provides the `df` utility to display the following information for each specified file system:

- File system name
- Total space allocated in the file system
- Amount of space allocated to existing files
- Total amount of space available for the creation of new files by unprivileged users
- Percentage of normally available space that is currently allocated to all files on the file system

To display the disk space information in human readable format, use the `df -h` command (as shown in the example in the slide). For more information about other options of the `df` command, refer to the `df`(1M) man page.

Displaying Information About Network Services

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

```
# svcs network/physical
STATE      STIME     FMRI
online     15:35:09  svc:/network/physical:upgrade
online     15:35:09  svc:/network/physical:default
```

ORACLE

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), which controls the operating system services. To see these services, use the `svcs` command. For more information about the `svcs` command, refer to the `svcs`(1) man page.

Note: SMF is discussed in the lesson titled "Administering Services by Using SMF."

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/v4           dhcp              ok        10.0.2.15/24
lo0/v6            static            ok        ::1/128
net0/v6           addrconf          ok        fe80::a00:27ff:fe4c:d1cb/10
```


Another network-related information that 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/v4` and `net0/v6`) and verify that these network interfaces are up and running. In the lesson titled “Administering the 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, by using the Live Media GUI installation option, the IP address was obtained automatically from DHCP by the reactive network.

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

Baseline System Information Commands: Summary

System Information	Command
Host name	`hostname`
Host ID	`hostid`
Basic system information	`uname -a`
Operating system release information	`cat /etc/release`
Disk configuration	`format`
Installed memory	`prtconf | grep Memory`
Disk space information	`df -h`
Network services information	`svcs network/physical`
Network interface information	`ipadm show-addr`

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

The table in this slide summarizes the commands used to display the 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. For further reading, you can also refer to http://docs.oracle.com/cd/E36784_01/html/E36819/sysressysinfo-75169.html#scrolltoc.

Quiz

The Oracle Device Detection Tool can be used to determine whether:

a. The OS is installed correctly
b. There are errors on the internal or external hard disks
c. A device driver is available

Answer: c

Quiz

Live Media can be installed only on x86 (64-bit only) hardware.

a. True

b. False

Answer: a

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

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 the following:
  - Basic system information
  - System release information
  - Boot disk configuration
  - Installed memory size
  - Network information

This practice should take you about 10 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Describe the Oracle Solaris 11 OS
- Implement a plan for an Oracle Solaris 11 OS installation
- Install the Oracle Solaris 11 OS by using the Live Media installer
- Install the Oracle Solaris 11 OS by using the text installer
- Verify the installed OS

3

Managing Boot and Shutdown of a System

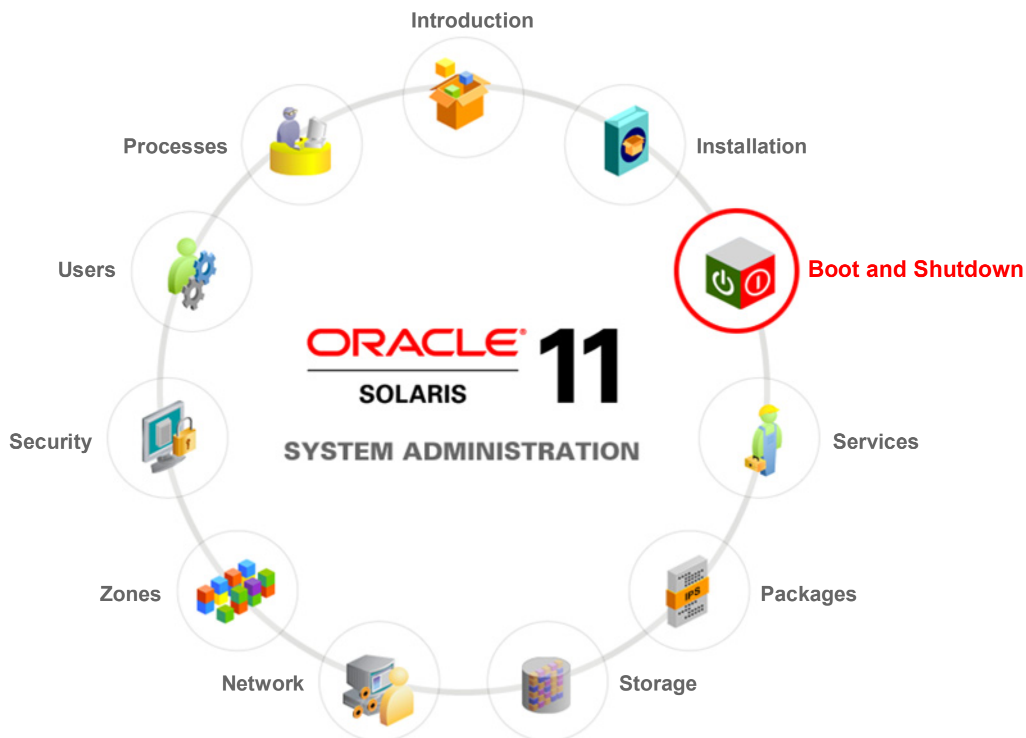
Objectives

After completing this lesson, you should be able to:

- Analyze the boot design and boot process
- Boot a SPARC-based system
- Boot an x86-based system
- Shut down a system

In this lesson, you are introduced to booting and shutting down x86 and SPARC systems.

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. Although Oracle Solaris 11 is designed to run continuously, there are times when the system must be rebooted or shut down. 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. If it is necessary to bring the system down for any reason, you need to know how to do it in a way that does not unexpectedly disrupt the services and resources that the system makes available to your end users.

Lesson Agenda

- **Analyzing the Boot Design and Boot Process**
- Booting a SPARC-Based System
- Booting an x86-Based System
- Shutting Down a System

Reasons to Shut Down and Boot a System

- Turning off system power due to anticipated power outage
- Changing kernel parameters in the `/etc/system` file
- Performing file system maintenance, such as backing up or restoring system data
- Repairing a system configuration file, such as `/etc/system`
- Adding or removing hardware from the system
- Booting a system for recovery purposes due to a lost `root` password, or to fix a file system or a similar problem
- **(x86 only)** Recovering from a problem with the GRUB configuration
- Recovering from a hung system by forcing a crash dump
- Booting the system to track down a system problem

Although Oracle Solaris 11 is designed to run continuously, there are times when the system must be rebooted or shut down. Having a basic understanding of the steps that the system performs when it boots and shuts down is a critical skill for all system administrators.

System startup requires an understanding of the hardware and operating system functions that are required to boot a system. This lesson discusses the operations that the system must perform from the time that you turn on power to the system until you receive a system login prompt. In addition, it covers the steps required to properly shut down a system.

Because of the difference in hardware, the kernel initialization phase of the boot process differs between the SPARC and x86 platforms.

Oracle Solaris Boot Architecture: Overview

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

Note: PROM stands for Programmable Read-Only Memory and GRUB stands for Grand Unified Bootloader.

All Oracle Solaris SPARC systems have resident boot PROM firmware that controls the operation of the system before the OS has been booted. All x86 systems use an open source boot loader called GRUB. Beginning with Oracle Solaris 11.1, GRUB was updated to GRUB 2. GRUB is responsible for loading a boot archive into the system's memory.

A boot archive is a subset of a root file system. This boot archive contains all the kernel modules and `driver.conf` files, in addition to a few configuration files. These files are located in the `/etc` directory. Files in the boot archive are read by the kernel before the `root` file system is mounted. After the `root` file system is mounted, the boot archive is discarded by the kernel from memory. Then, file I/O is performed against the `root` device.

The `bootadm` command manages the boot archive on both SPARC and x86 platforms, including details related to boot archive update and verification. During a normal system shutdown, the shutdown process compares the boot archive's contents with the root file system. If there have been updates to the system, such as drivers or configuration files, the boot archive is rebuilt to include these changes so that on reboot, the boot archive and the root file system are synchronized.

The files that are part of the x86 boot archive are located in the `/platform/i86pc/amd64/archive_cache` directory. The files in the SPARC boot archive are located in the `/platform/'uname -m'/archive_cache` directory. To list the contents of the boot archive on both SPARC and x86 platforms, use the `bootadm list-archive` command:

```
$ bootadm list-archive
```

If any files in the boot archive are updated, the archive must be rebuilt. The `bootadm update-archive` command enables you to manually rebuild the boot archive. The command can be used either as a preventative measure or as part of a recovery process.

```
# bootadm update-archive
```

For more information about the boot architecture, refer to http://docs.oracle.com/cd/E36784_01/html/E36801/gktkp.html#scrolltoc.

Boot PROM for SPARC Systems

The boot PROM firmware:

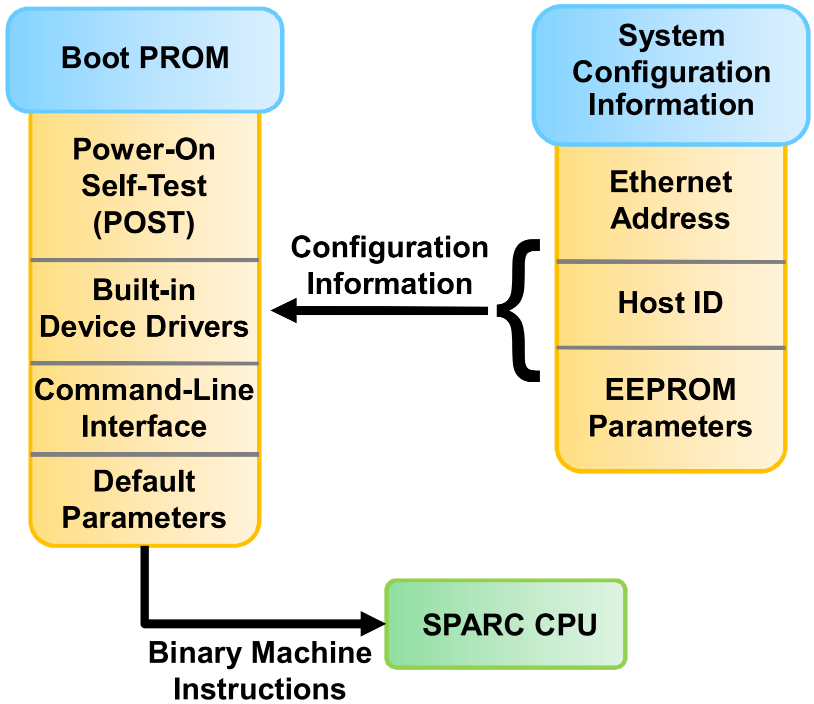
- Provides basic hardware testing and initialization before loading the operating system
- Enables booting from a wide range of devices
- Provides a user interface
- Has access to a standard set of device drivers

In addition to booting the operating system, the OpenBoot architecture enables the boot PROM firmware to perform 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.

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.

Bootstrapping Process for SPARC Systems
(Boot PROM Initialization)



The process of loading and executing a stand-alone program is called *bootstrapping*. Typically, the stand-alone program is the operating system kernel. However, any stand-alone program can be booted instead of the kernel.

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. As part of this process, system configuration information is provided to the boot PROM. 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.
  Note: The user-configurable parameters are known as Non-Volatile Random Access Memory (NVRAM) variables or Electrically Erasable Programmable Read-Only Memory (EEPROM) parameters. They allow an administrator to control things such as the default boot device, the level of POST, and so on.
- Installs the console and displays the `ok` prompt. After the boot PROM initializes the system, the banner is displayed on the console. The system checks the parameters stored in the boot PROM and NVRAM to determine whether and how to boot the OS.

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.

You can display and modify the value of parameters in EEPROM from the OS by using the `eeprom` command. You do not need any special privileges to display EEPROM parameters. However, to modify these parameters, you must assume the `root` role.

On x86 platforms, the setting of EEPROM properties is simulated by:

- Storing Oracle Solaris–specific properties in the `/boot/solaris/bootenv.rc` file
- Manipulating the GRUB menu to simulate the effect of setting certain EEPROM properties
- Implementing NVRAM storage for variables specific to the UEFI environment

For UEFI-enabled systems, the EEPROM parameters are stored in two places. Oracle Solaris–specific variables are stored in the `bootenv.rc` file. UEFI-specific variables are set in the NVRAM store. Unlike SPARC with OpenBoot PROM, Oracle Solaris variables are not consumed by UEFI firmware. To make that the UEFI-specific variables are available, use the `-u` option with the `eeprom` command. The `-u` option is supported starting with the Oracle Solaris 11.2 release.

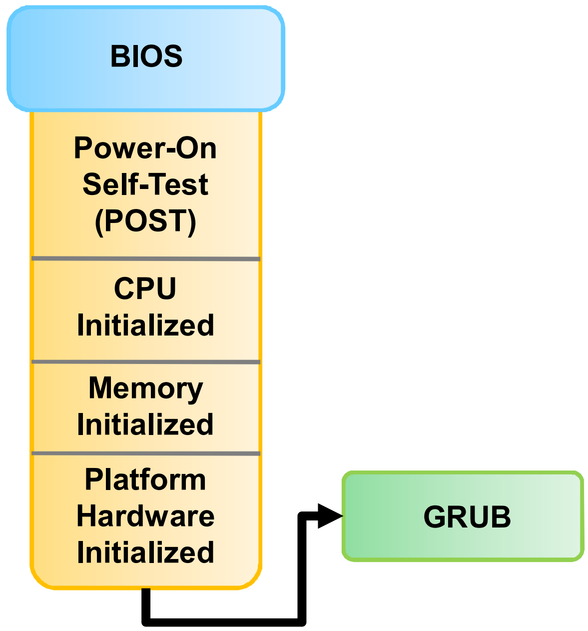
Most UEFI variables are in binary format, and are translated to a readable format. When translation is not possible, a hexdump is printed.

The following example shows how to display all UEFI parameters on a system in UEFI mode.

```
# eeprom -u
MonotonicCounter=0x1f2
OsaBootOptNum=0xffff
ConOut=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147549
440)
/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(None
)/VenPcAnsi()
ConIn=/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtr
l(None)/VenPcAnsi()
/PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x8,0x0)
BootOrder=Boot0000 Boot0001 Boot0002 Boot0003 Boot0004 Boot0005 Boot0006
Lang=eng
```

```
PlatformLang=en-US

Timeout=0x1

Boot0001=description:string=[UEFI]USB:USBIN:USB USB Hard Drive ,
flags:int=1, device_path:
string=/PciRoot(0x0)/Pci(0x1a,0x0)/USB(0x1,0x0)/USB(0x2,0x0)/HD(1,MBR,0x004D
5353,0x800,0x3b5800), optional_data:string=AMBO

Boot0002=description:string=[UEFI]PXE:NET0:Intel(R) Ethernet Controller 10
Gigabit X540-AT2, flags:int=1,
device_path:string=/PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x0)/MAC(002128e77478)
, optional_data:string=AMBO

Boot0003=description:string=[UEFI]PXE:NET1:Intel(R) Ethernet Controller 10
Gigabit X540-AT2, flags:int=1,
device_path:string=/PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x1)/MAC(002128e77479)
, optional_data:string=AMBO

Boot0004=description:string=[UEFI]PXE:NET2:Intel(R) Ethernet Controller 10
Gigabit X540-AT2, flags:int=1,
device_path:string=/PciRoot(0x1)/Pci(0x1,0x0)/Pci(0x0,0x0)/MAC(002128e7747a)
, optional_data:string=AMBO

Boot0005=description:string=[UEFI]PXE:NET3:Intel(R) Ethernet Controller 10
Gigabit X540-AT2, flags:int=1,
device_path:string=/PciRoot(0x1)/Pci(0x1,0x0)/Pci(0x0,0x1)/MAC(002128e7747b)
, optional_data:string=AMBO

Boot0006=description:string=[UEFI]SAS:PCIE3:ATA HITACHI HDS7225SA81A,
flags:int=1, device_path:string=/PciRoot(0x0)/Pci(0x3,0x0)/Pci(0x0,0x0) \

/MessagingPath(10,2c00b .... 12010100) /HD(1,GPT,BCB01265-4665-F1CA-8BF5-
9C4FB95962FA,0x100,0x80000), optional_data:string=AMBO

Boot0000=description:string=Oracle Solaris s12_13, flags:int=1, device_path:
string=HD(1,GPT,C7398875-60D2-A9E0-83EE-94DAA21B0383,0x100,0x80000),
file_path:string=/EFI/Oracle/grubx64.efi

USB_POINT=5139417f00000000

ConOutDev=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147
549440)/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCt
rl(None)/VenPcAnsi()

ConInDev=/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlow
Ctrl(None)/VenPcAnsi()/PciRoot(0x0)/Pci(0x1d,0x0)/USB(0x1,0x0)/USB(0x8,0x0)

BootOptionSupport=0x1
ErrOutDev=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147
549440)/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCt
rl(None)/VenPcAnsi()

ErrOut=/PciRoot(0x0)/Pci(0x1c,0x7)/Pci(0x0,0x0)/Pci(0x0,0x0)/AcpiAdr(2147549
440)/PciRoot(0x0)/Pci(0x1f,0x0)/Serial(0x0)/Uart(115200,8,N,1)/UartFlowCtrl(
None)/VenPcAnsi()

PlatformLangCodes=en-US

S3PerfAdd=hexdump:989fd6aa00000000

LangCodes=eng

BootCurrent=Boot0000
```

Bootstrapping Process for x86 Systems
(BIOS and GRUB Initialization)



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 to choose from. A boot entry is an OS instance that is installed in your system. GRUB also has a command-line interface that is accessible from the menu interface for performing various boot commands.

GRUB 2

- GRUB 2 uses an entirely different configuration than GRUB Legacy.
- It is managed through the `grub.cfg` file.
- Unlike the `menu.lst` file used in GRUB Legacy, `grub.cfg` file should never be edited.
- GRUB 2 supports the Unified Extensible Firmware Interface (UEFI) and the GUID Partition Table (GPT) partitioning scheme.

In the Oracle Solaris 11.1 OS, GRUB 2 is added as the default x86 boot loader. GRUB 2 is a powerful and more modular boot loader that supports a wider range of platforms and firmware types, including booting from the BIOS and Unified Extensible Firmware Interface (UEFI) firmware, and booting from GUID Partition Table (GPT) partitioned disks of any size, which not only supports UEFI-specified, GPT partitioning schemes, but also disks that are larger than 2 TB.

Note: UEFI is a specification that defines a software interface between an operating system and a platform firmware. Oracle Solaris supports x86-based systems with the 64-bit UEFI firmware. Installation on the UEFI firmware is supported through the DVD, USB, and network installation methods. UEFI version 2.1+ is required.

The GRUB Legacy system configuration is managed through the `menu.lst` file, but GRUB 2 does not use a `menu.lst` file. Instead, GRUB 2 uses a configuration file, `grub.cfg`, to store the same type of information. Similar to the `menu.lst` file, the `grub.cfg` file is located at the top level of the ZFS dataset for the root pool, */pool-name/boot/grub*—for example, `/rpool/boot/grub/grub.cfg`. The syntax of the `grub.cfg` file is based on a subset of `bash` scripting, which is more complex and powerful than the directive-like language that is used in the `menu.lst` file. Unlike the `menu.lst` file, the `grub.cfg` file is automatically regenerated by boot management commands. Therefore, this file should never be directly edited because any edits are immediately destroyed when the `grub.cfg` file is regenerated.

For information about GRUB 2 and how to upgrade your GRUB Legacy system to a release that supports GRUB 2, refer to the http://docs.oracle.com/cd/E36784_01/html/E36801/gkvif.html#scrolltoc and http://docs.oracle.com/cd/E36784_01/html/E36801/gluae.html#scrolltoc URLs.

For information about GRUB 2 Partition and Device Naming Scheme, refer to http://docs.oracle.com/cd/E36784_01/html/E36801/gkvii.html#scrolltoc.

Boot Process

Boot Loader Phase	The root file system archive is loaded.
Booter Phase	The boot archive is read and executed.
Ramdisk Phase	The kernel image is extracted and executed.
Kernel Phase	The OS is initialized and the root file system is mounted.
`init` **Phase**	The `init` daemon starts the `svc.startd` daemon.
`svc.startd` **Phase**	The `svc.startd` daemon starts the 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: This is the only phase of the boot process that requires knowledge of the root file system format. 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 are 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.

Verified Boot

Starting from Oracle Solaris 11.2 release, you can use the Verified Boot feature to secure a system's boot process. The feature protects the system from threats such as the following:

- Corruption of kernel modules
- Insertion or substitution of malicious programs that masquerade as legitimate kernel modules, such as Trojan viruses, spyware, and root kits
- Installation of unauthorized third-party kernel modules

Malicious programs can pass information to third parties as well as alter the behavior of the Oracle Solaris OS. Although third-party modules are typically non-malicious, they might violate policies that control site changes. Therefore, the system also needs protection from unauthorized installation of these modules.

For more information about the Verified Boot feature, refer to http://docs.oracle.com/cd/E36784_01/html/E37121/gmwce.html#scrolltoc and http://docs.oracle.com/cd/E36784_01/html/E37121/gmwmn.html#scrolltoc.

SMF and Booting

In Oracle Solaris 11, SMF provides an infrastructure that augments the traditional UNIX `startup` scripts, `init` states, and configuration files.

- The boot process has become faster with multiple SMF services started simultaneously.
- Due to SMF, the boot process creates fewer messages now.
- The SMF services do not display a message by default when they are started during the boot process.
- All information that was provided by the boot messages can now be found in a log file for each SMF service under `/var/svc/log`.
- You can use the `svcs` command to diagnose boot problems.
- To generate a message when each service is started during the boot process, you use the `-v` option with the `boot` command.

Note: The Service Management Facility (SMF) is a software framework that is responsible for managing services on a system—whether they are critical system services essential to the working of the system or application services, such as a database or web server. You learn more about the Service Management Facility (SMF) feature in the next lesson titled "Managing Services by Using SMF."

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 the root file system are synchronized.

Oracle Solaris boot archive updates and verification on both SPARC and x86 platforms are handled automatically by the `bootadm` command. During an installation or upgrade, the `bootadm` command creates an initial boot archive. During 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 the reboot, the boot archive and the root file system are synchronized.

Note: The Service Management Facility (SMF) feature, about which you will learn in the next lesson titled "Managing Services by Using SMF," manages the following boot archive services:

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

Fast Reboot

- Is supported on both SPARC and x86 platforms
- 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 make the system boot time significantly faster by allowing certain boot loader steps and firmware checks to be bypassed and to thus minimize system down time. 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 SMF and implemented through a boot configuration service, `svc:/system/boot-config`, which is enabled by default on x86 systems. The `boot-config` service provides a way to set or change default boot configuration parameters. Administrators can enable Fast Reboot on SPARC by modifying the property as follows:

```
# svccfg -s system/boot-config:default setprop \
config/fastreboot_default=true
# svcadm refresh system/boot-config:default
```

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 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.

If you are interested in understanding further about how the Fast Reboot feature works, refer to the http://download.oracle.com/otndocs/tech/OTN_Demos/x86/x86-OTN-Demo/x86-OTN-Demo.html demonstration and http://docs.oracle.com/cd/E36784_01/html/E36801/ghsbc.html#scrolltoc.

SMF Milestones

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

A state of “init” defines what services and resources are available to users. A system can be in only one init state at a time. The table in the slide lists all init states.

Milestones are checkpoints in the operating system. Milestones are services that do not run any applications but do have a dependent list of services. After those services are online, a milestone is marked online. A milestone ensures that an expected group of services is up and running, so you do not have to check each individual service.

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 course to administer services.

For more details on the init states, refer to http://docs.oracle.com/cd/E36784_01/html/E36801/runlevels-130.html#scrolltoc.

Quiz

Which SMF service helps in implementing the Fast Reboot feature?

a. `svc:/system/boot-update:default`
b. `svc:/system/boot-archive:default`
c. `svc:/system/boot-archive-update:default`
d. `svc:/system/boot-config:default`
e. `svc:/system/boot-config-update:default`

Answer: d

Quiz

In which phase of the boot process is the OS initialized and a minimal root file system mounted on the RAM disk that was constructed from the boot archive?

a. Kernel phase
b. Boot loader phase
c. `svc.startd` phase
d. Booter phase
e. Ramdisk phase
f. `init` phase

Answer: a

Lesson Agenda

- Analyzing the Boot Design and Boot Process
- **Booting a SPARC-Based System**
- Booting an x86-Based System
- Shutting Down a System

ORACLE

Booting a SPARC-Based System

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

Booting a SPARC System to Multiuser-Server Milestone (`init` State `3`)

1. Boot the system to the multiuser-server milestone 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 the multiuser-server milestone.

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


When you boot a system to a multiuser state:

1. The `init` process is started and reads the properties defined in the `svc:/system/environment:init` SMF service to set any environment variables. By default, only the `TIMEZONE` variable is set.
2. `init` then reads the `/etc/inittab` file and performs the following:
   a. Executes any process entries that have `sysinit` in the action field so that any special initializations can take place before users log in to the system
   b. Passes the startup activities to `svc.startd`

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 the multiuser-server milestone. The assumption is that the operating system is at init state 0 (power-down state).

To boot a system to the multiuser-server milestone, 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 the multiuser milestone.

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

Booting a SPARC System to Single-User Milestone (`init` State S)

- Boot the system to the single-user milestone by using the following command at the `ok` prompt:

```
ok boot -m milestone=single-user
```

- 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 the single-user milestone.

```
# who -r
. run-level S Nov 11 10:15 S 0 S
#
```


To boot a system to the single-user milestone, perform the steps shown in the slide. The 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 the single-user milestone. When you are done with your maintenance task, you must press Ctrl + D to bring the system to the multiuser milestone (init state `3`, which is the default state).

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.

Using the Basic Boot PROM Commands

Boot PROM Command	Description
`banner`	Displays a system's PROM revision number
`setenv`	Sets the specified NVRAM parameter to a value For example, `ok setenv auto-boot? false` sets the PROM `auto-boot?` value to `false`.
`reset-all`	Clears system registers and resets the entire system
`sifting probe`	Displays the probe commands that are available on your system
`probe-`*`device`*	Identifies the devices on the system
`devalias`	Identifies the device aliases and the associated paths of devices that *might* be connected to the system
`printenv`	Displays all current and default parameter values
`eeprom`	Helps in displaying and modifying the value of parameters in EEPROM

Practice 3-1 Overview: Booting and Shutting Down a SPARC Host

In this practice, you are given a demonstration of how to use some of the `bootprom` commands on a SPARC-based host machine.

This practice should take about five minutes to complete.

Lesson Agenda

- Analyzing the Boot Design and Boot Process
- Booting a SPARC-Based System
- **Booting an x86-Based System**
- Shutting Down a System

Booting an x86 System

- Booting an x86 system to the multiuser-server milestone
- Booting an x86 system to the single-user milestone

Booting an x86 System to Multiuser-Server Milestone

1. Reboot the system by using the `reboot` command.

```
# reboot -p
```

2. When the GRUB menu appears, press Enter to boot the default OS instance to the multiuser-server milestone.
3. When the login prompt appears, log in to the system as `root`.
4. Verify that the system has booted to the multiuser-server milestone (init state `3`).

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


To boot a system that is currently in a power-down state to the multiuser-server milestone (init state `3`), 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 the multiuser-server milestone. The login prompt is displayed when the boot process has finished successfully.

Note: The SMF service, `svc:/system/boot-config:default`, is enabled by default with the `config/fastreboot_default` property set to `true`, thereby enabling the Fast Reboot feature. On a system where the Fast Reboot feature is enabled, if you run the `init 6` command to reboot the system, it will bypass certain firmware initialization steps and will not display the GRUB menu during reboot. Therefore, you use the `-p` option with the reboot command to disable the Fast Reboot feature.

Booting an x86 System to Single-User Milestone (init State S)

1. Reboot the system by using the `reboot -p` 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, enter `-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 the single-user milestone.

ORACLE

To boot a system to the single-user milestone, perform the steps shown in the slide. The 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, refer to 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 the single-user milestone. When you are done with your maintenance task, you must reboot the system.

Initiating Fast Reboot on 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.

Using the `bootadm` Command

`bootadm` Sub-command	Description
`list-menu`	Lists the contents of the `grub.cfg` file
`generate-menu`	Generates the `grub.cfg` file
`set-menu`	Sets a particular GRUB menu entry as the default and other menu options and boot loader options; helps in maintaining the GRUB menu
`add-entry`	Adds a boot entry to the GRUB menu
`change-entry`	Changes the attributes of a specified boot entry in the GRUB menu
`install-bootloader`	Installs the system boot loader **Note:** This subcommand applies to both x86 and SPARC platforms.
`remove-entry`	Removes a boot entry from the GRUB menu

Note: The `bootadm` command helps you to manage boot configurations on x86 platforms. For examples on how to use the commands listed in the slide, refer to http://docs.oracle.com/cd/E36784_01/html/E36801/gkvhz.html#scrolltoc.

Practice 3-2 Overview: Booting an x86/64 Host

In this practice, you perform a hands-on exercise on how to boot an x86/64 host system.

This practice should take about 10 minutes to complete.

Lesson Agenda

- Analyzing the Boot Design and Boot Process
- Booting a SPARC-Based System
- Booting an x86-Based System
- **Shutting Down a System**

Shutting Down a System

- Shutting down a server:
  - The `shutdown` command is used.
  - A clean shutdown is performed.
  - Superuser privileges are required.
- Shutting down a stand-alone system:
  - The `init` command is used.
  - A 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. Some system administration tasks and emergency situations such as the following require that the system be shut down to a level where it is safe to disrupt power.

- Adding or removing hardware
- Preparing for an expected power outage
- Performing file system maintenance, such as a backup
- Bringing a system 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 section, 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 `root` role privileges to execute either of these commands.

Keep the following in mind when you shut down a system:

- Use either the `shutdown` or the `init` command to shut down a system. Both commands perform a clean system shutdown, which means all system processes and services are terminated normally.
- You need to be in the `root` role to use the `shutdown` and `init` commands.
- Both the `shutdown` and `init` commands take a run level as an argument.
- The three most common run levels are as follows:
  - **`init 3`:** All system resources are available and users can log in. By default, booting a system brings it to the multiuser-server milestone (with NFS resources shared), which is used for normal day-to-day operations.
  - **`init 6`:** The system is shut down to the power-down state, and then the system is rebooted to a multiuser level with SMB or NFS resources shared.
  - **`init 0`:** The operating system is shut down and it is safe to turn off power. You need to bring a system to power-down state whenever you move the system, or add or remove hardware.

Determining Who Is Logged In to a System

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

```
$ who
holly     console  Nov 11 07:30
kryten    pts/0    Nov 11 07:35 (starlite)
lister    pts/1    Nov 11 07:40 (bluemidget)
```

Best practice: Always send an additional email notification to logged in 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 the example in the slide.

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 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 init state that you specified.
5. When you have completed your administration tasks, press Ctrl + D to return to the default system init state.
6. Verify that the system is at the init state that you specified in the `shutdown` command.

Specified init State	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:`

When shutting down a server that is operating at the 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 the single-user milestone 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`. `init` states `0` and `5` are reserved states for shutting the system down. `init 5` goes a step further and powers down the machine. `init 6` reboots the system. `init 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 init state, you can perform the system administration tasks that required you to change the system's init state.

Note for step 6: The table in the slide shows the prompts that you see (by system type) based on the init state you specified in the `shutdown` command. For example, if you specified an init state 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 init state `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 init state `S`, run `init S`.

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

ORACLE

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

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

Practice 3-4 Overview: Shutting Down an x86/64 Host

In this practice, you perform a hands-on exercise to learn how to shut down an x86/64-based system.

This practice should take you about 10 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Analyze the boot design and boot process
- Boot a SPARC-based system
- Boot an x86-based system
- Shut down a system

4

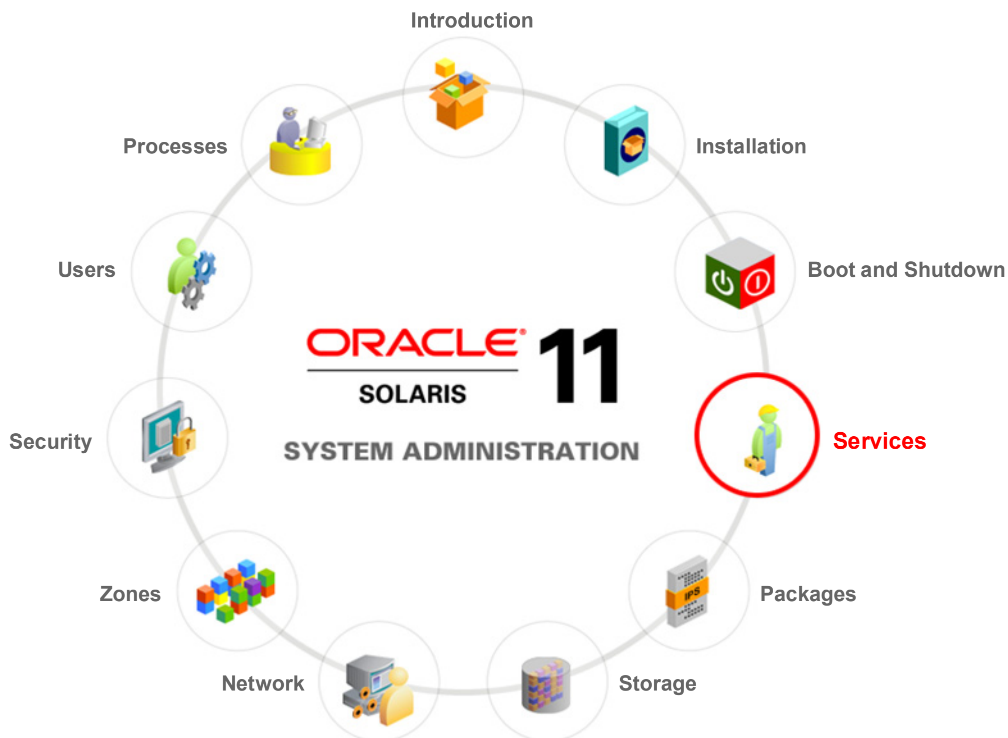
Administering Services by Using SMF

Objectives

After completing this lesson, you should be able to:

- Explain the SMF feature and its components
- Administer SMF services
- Manage SMF services by using the graphical user interface

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 learned about the boot and shutdown processes. Now, you turn your attention to the services that are responsible for keeping the system running day after day.

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

Lesson Agenda

- **Describing SMF and Its Components**
- Administering SMF Services
- Managing SMF Services by Using the GUI

Importance of Services Administration

Services administration is required to ensure that:

- System and application services run smoothly and efficiently
- Systems continue to be available by providing all essential services even during a system failure

Your company's Oracle Solaris 11 implementation plan may contain several subplans that include maintenance and contingency planning, a part of which is achieved by services administration. The focus of services administration planning is to:

- Ensure that the default services that run on the system are running smoothly and efficiently
- Improve the availability of a system by ensuring that essential system and application services run continuously even in the event of hardware or software failures

Service Management Facility

- Provides a framework for managing:
  - System and application services
  - Interaction of services with other dependent 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 an individual log file for each service

The Service Management Facility (SMF), first introduced in Oracle Solaris 10, is a software framework that is responsible for managing services on a system—whether they are critical system services essential to the working of the system or application services, such as a database or web server.

The framework infrastructure augments the traditional UNIX startup scripts, `init` run levels, and configuration files. SMF provides a mechanism to define relationships between applications or services, so that dependent services can automatically be restarted when necessary. The information that is required to manage each service is stored in the service configuration repository, which provides a simplified way to manage each service.

You can invoke a set of actions on a service that is defined by SMF. These actions, which include enable, disable, refresh, restart, and mark, can be manually manipulated by using the `svcadm` command. Each service is managed by a service restarter, which carries out the administrative actions.

SMF Capabilities

- Booting faster
- Restarting failed services
- Inspecting services
- Managing services
- Configuring services
- Auditing service changes
- Securely delegating tasks
- Creating new services
- Debugging service problems
- Configuring failure notification
- Converting `inetd.conf` configurations to SMF services
- Converting SMF service properties to configuration files

The SMF framework is always active on an Oracle Solaris 11 system. SMF provides the following capabilities:

- **Boot faster:** SMF speeds booting of large systems by starting services in parallel according to the dependencies of the services.
- **Restart failed services:** SMF services have well-defined dependency relationships with other services. If a service fails, SMF reports any affected dependent services and automatically attempts to restart failed services in dependency order.
- **Inspect services:** View relationships between services and processes and view the values of service properties.
- **Manage services:** Enable, disable, and restart services. These changes can persist through upgrades and reboots, or you can specify temporary changes.
- **Configure services:**
  - Change the values of service properties.
  - Add and delete custom properties.
- **Audit service changes:** SMF writes Oracle Solaris audit records for every administrative change to a service or its properties. SMF can show whether a property value or service state was set by an administrator.

- **Securely delegate tasks:** SMF securely delegates tasks to non-root users, including the ability to modify properties and enable, disable, or restart services.
- **Create new services:** Easily create a new instance of an existing service, copy and modify an existing service, or use a service creation tool.
- **Debug service problems:** Easily display an explanation for why an enabled service is not running or why a service is preventing another service from running.
- **Configure failure notifications:** SMF allows you to configure how you want to be notified of particular software events or hardware faults.
- **Convert `inetd.conf` configurations to SMF services.**
- **Convert SMF service properties to configuration files:** This mechanism provides a bridge for services that are managed by SMF but interact with applications that still require configuration files.

SMF 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)
- 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.

The SMF service has a well-defined state (enabled, disabled, offline, or maintenance) and usually has a relationship with the other dependent services that are required to be running on the system first. This provides a key benefit in that services can be started in parallel during system startup, resulting in a much faster boot when compared to the legacy `init` framework, which is able to start processes only in sequence and must wait until they complete.

Each service is a service manifest that describes some basic information about the service, the service dependencies that are required, any required service configuration, and how SMF should start and stop the service. A service, after it is started, can start several different processes that are tied together as part of a service contract. This means that you need to manage only the higher-level service, rather than worrying about a series of individual processes and the start order that might be required by those processes. If a service fails for any reason, whether during a hardware or software fault, SMF automatically detects the failure and restarts the service and any dependent services. Examples of service names are `login`, `SSH server`, and `hostid`.

The fundamental unit of services administration in the SMF framework is the *service instance*. Each SMF service could be running multiple times on a system accounting for slightly different configurations. These different configurations are called service instances. Each *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`.

Service Instance

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 points to the system facilities (`system`)
- Within `system`, the next level category is `filesystem`
- The next lower category is `root`, which points to the `root` file system
- The service name is `system/filesystem/root:default`
- An instance of the service is `default`

SMF identifies each instance of a service by a service identifier. This service identifier is in the form of a Fault Management Resource Identifier (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 names include prefixes to help identify the purpose of each service. These prefixes include names such as `application`, `device`, `milestone`, `network`, or `system`. The `site` prefix is reserved for site-specific customizations, which means that a service named `svc:/site/`*`service-name`* will never conflict with the services delivered in an Oracle Solaris release.

Service Models

SMF services are one of the following three models:

SMF Model	Description
Transient service	The service performs some task, and then exits without starting any long running processes.
Child or wait service	The service is restarted whenever its child process exits cleanly. A child process that exits cleanly is not treated as an error.
Contract or daemon service	The service starts a long-running daemon or starts several related processes that are tied together as part of a *service contract*. The contract service manages the processes that it starts and any dependent services and their start order. You only need to manage the high-level service.

Service States

A service instance can have different states, as listed in the following table:

Service State	Description
`online`	Enabled and successfully started
`offline`	Enabled but not yet running or available to run
`offline*`	Process that is in the state of starting
`disabled`	Not enabled and not running
`legacy_run`	Running. The service is not directly managed by SMF, but it was started at some point.
`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 limited capacity

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

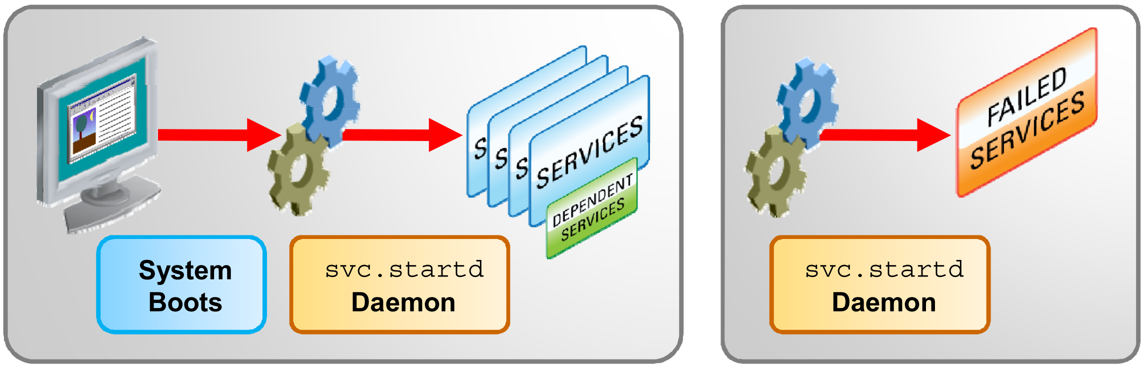
The service configuration repository stores persistent configuration information, as well as SMF runtime data for services. The repository is distributed among local memory and local disk-based files, and is called `/etc/svc/repository.db`. The service configuration repository can be manipulated or queried only by using SMF interfaces. For more information about manipulating and accessing the repository, refer to the `svccfg`(1M) and `svcprop`(1) man pages. You can learn more about the service configuration repository, how to create your own service configurations, and how to fix a corrupted repository in the follow-on course titled *Oracle Solaris 11 Advanced Administration*.

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 view 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 that 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, `svc.startd` retrieves the information in the service configuration repository. Next, the daemon starts services when their dependencies are met. The daemon is also responsible for restarting services that have failed and for shutting down services whose dependencies are no longer satisfied. The daemon uses operating system events, such as process death, to keep track of service states.

Note: The `svc.startd` daemon is also responsible for ensuring that the system boots to the appropriate milestone. Recall from lesson 3 titled "Managing Boot and Shutdown of Systems" that 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, `svc.startd` boots to the built-in milestone `all`, which includes all the system-enabled services.

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

Quiz

If a service instance is in the state of starting, which of the following states will it be identified with?

a. `uninitialized`

b. `online`

c. `offline`

d. `offline*`

Answer: d

Lesson Agenda

- Describing SMF and Its Components
- **Administering SMF Services**
- Managing SMF Services by Using the GUI

Administering SMF Services

- Monitoring services with `svcs`
  - Listing services information
  - Displaying the status of a service
  - Displaying the service dependents
  - Displaying the dependencies of a service
- Administering services with `svcadm`
  - Disabling a service
  - Enabling a service
  - Restarting a service
- Setting up service state transition notifications
  - Installing the `smtp-notify` package
  - Enabling the `smtp-notify:default` service
  - Configuring service state transition notifications
    - Service state transition notification: Example
  - Managing service state transition notifications

When a system is started, SMF starts up several 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
- Setting up service state transition notifications by using the `smtp-notify` service

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

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

Listing Services Information

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

```
# svcs
STATE          STIME    FMRI
legacy_run     1:25:08 lrc:/etc/rc2_d/S47pppd
legacy.run     1:25:08 lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run     1:25:08 lrc:/etc/rc2_d/S89PRESERVE
<output omitted>
```

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

```
# svcs -a
STATE          STIME    FMRI
legacy_run     1:25:08 lrc:/etc/rc2_d/S47pppd
legacy.run     1:25:08 lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run     1:25:08 lrc:/etc/rc2_d/S89PRESERVE
disabled       1:23:38 svc:/system/device/mpxio-upgrade:default
<output omitted>
```


To see the services that are currently running in the system (including services that have been disabled temporarily), use the `svcs` command with no arguments, as shown in the first partial example in the slide. 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 in the system, including the disabled ones and incomplete ones, use the `svcs -a` command, as shown in the second partial example in the slide. The incomplete services can be further viewed by using `svcs -x` *`FMRI`*.

Displaying the Status of a Service Instance

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

```
# svcs -l svc:/network/ssh:default
fmri          svc:/network/ssh:default
name          SSH server
enabled       true
state         online
next_state    none
state_time    July 31, 2014 09:35:56 PM MDT
logfile       /var/svc/log/network-ssh:default.log
restarter     svc:/system/svc/restarter:default
contract_id   110
manifest      /etc/svc/profile/generic.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)
```

ORACLE

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 in the slide. 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. For more details about the `svcs` command and its options, refer to the `svcs`(1) man page.

In the slide 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.

Displaying the Service Dependents

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

```
# svcs -D svc:/network/ssh:default
STATE          STIME    FMRI
online         1:25:05  svc:/milestone/self-assembly-complete:default
online         1:25:09  svc:/milestone/multi-user-server:default
```


Sometimes, services depend on each other 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 the service instances that depend on another service, use the `svcs -D` command followed by the service identifier, as shown in the example in the slide.

Displaying the Dependencies of a Service

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

```
# svcs -d svc:/network/ssh:default
STATE          STIME    FMRI
disabled       1:23:51  svc:/network/ipfilter:default
online         1:24:04  svc:/system/cryptosvc:default
online         1:24:09  svc:/network/loopback:default
online         1:24:11  svc:/system/utmp:default
online         1:24:28  svc:/network/physical:default
online         1:24:36  svc:/system/filesystem/local:default
online         1:25:04  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 in the slide.

Disabling a Service

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

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

3. Use the `svcs -l` *`FMRI`* command 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>
```

ORACLE

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 persists across reboots. This means that the only way to get the service running again is to enable it.

Disabling a Service Instance

The restarter for the service attempts to bring the specified instance to the `disabled` state. An instance can be permanently or temporarily disabled. A permanent “disabled” state is persistent across system reboot, and is the default. A temporary “disabled” state lasts only until reboot.

- To permanently disable the instance, use the `svcadm disable` *`FMRI`* command.
- To temporarily disable the instance, use the `-t` option, as `svcadm disable -t` *`FMRI`*. With this option, the service returns to an online state on reboot.
- To synchronously disable the instance, specify the `-s` option. The `svcadm` command will disable the instance and wait for the instance to enter the disabled state before returning. The `svcadm` command returns when the instance reaches the disabled state or when it determines that the instance requires administrator intervention to reach the `disabled` state. Use the `-T` option with the `-s` option to specify an upper bound in seconds to make the transition or determine that the transition cannot be made, for example, `svcadm disable -sT 10` *`FMRI`*.

Enabling a Service

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

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

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

```
# svcs -x svc:/network/ssh:default
svc:/network/ssh:default (SSH server)
 State: online since July 31, 2014 09:35:56 PM MDT
   See: sshd(1M)
   See: /var/svc/log/network-ssh:default.log
Impact: None.
```

ORACLE

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, you 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.

The restarter for the service attempts to bring the specified instance to the `online` state. An instance can be permanently or temporarily enabled. The permanent “enabled” state is persistent across system reboot, and is the default. The temporary “enabled” state lasts only until reboot.

- To permanently enable the instance, use the `svcadm enable` *`FMRI`* command.
- To temporarily enable the instance, use the `-t` option to specify a temporary “enabled” state, as `svcadm enable -t` *`FMRI`*.

- To synchronously enable the instance, specify the `-s` option. The `svcadm` command will enable the instance and wait for the instance to enter the online or degraded state before returning. The `svcadm` command returns when the instance reaches an online state or when it determines that the instance requires administrator intervention to reach an online state. Use the `-T` option with the `-s` option to specify an upper bound in seconds to make the transition or determine that the transition cannot be made, for example, `svcadm disable -sT 10` *`FMRI`*.

Refreshing and Restarting a Service

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

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

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

```
# 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, whereas others require a restart.

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

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

Note: When a refresh is performed, the running snapshot is taken or 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.

Restoring a Service That Is in Maintenance State

1. Use the `svcs -x` *`FMRI`* command to determine why the service is in maintenance state.
2. Use the `svcs -l` *`FMRI`* command to determine if any process that is dependent to the service has not stopped.
3. Use the `svcs -o CTID` *`FMRI`* command to obtain the contract ID of the service that you want to restore.
4. Use `pkill -9 -c` *`CTID`* to kill any remaining processes.
5. Use `svcadm clear` *`FMRI`* to restore the service.

If any service enters maintenance state due to some system error or due to human error, and requires the administrator's intervention to restore it, use the steps provided in the slide. In the following example, you can observe how the service `svc:/system/ocm:default` that is in maintenance state is being restored:

```
# svcs -a
STATE            STIME     FMRI
legacy_run       14:23:13  lrc:/etc/rc2_d/S47pppd
legacy_run       14:23:13  lrc:/etc/rc2_d/S89PRESERVE
disabled         14:21:48  svc:/system/device/mpxio-upgrade:default
...
...
online           11:11:18  svc:/network/ssh:deafult
maintenance      14:23:56  svc:/system/ocm:default
```

```
# svcs -x svc:/system/ocm:default
svc:/system/ocm:default (Oracle Configuration Manager (OCM))
 State: maintenance since November 11, 2013 02:23:56 PM UTC
Reason: Method failed.
   See: http://support.oracle.com/msg/SMF-8000-8Q
   See: http://download.oracle.com/docs/cd/E23562_01/doc.1035/e22050/tco.htm
   See: configCCR(1M)
Impact: This service is not running.
```

Note: Normally, when a service instance is in maintenance state, all processes associated with that instance have stopped. However, you should make sure before you proceed. The following command lists the contract ID for the selected FMRI.

```
# svcs -l svc:/system/ocm:default
fmri         svc:/system/ocm:default
name         Oracle Configuration Manager (OCM)
enabled      false (temporary)
state        maintenance
next_state   none
state_time   November 11, 2013 02:23:56 PM UTC
logfile      /var/svc/log/system-ocm:default.log
restarter    svc:/system/svc/restarter:default
contract_id  209
manifest     /etc/svc/profile/site/sc_profile.xml
manifest     /etc/svc/profile/generic.xml
dependency   require_all/none svc:/milestone/multi-user-server:default
   (online)
dependency   require_all/error svc:/milestone/network:default (online)
Dependency   require_all/none svc:/system.cryptosvc (online)
# svcs -o CTID svc:/system/ocm:default
CTID
              129
# pkill -9 -c 129
```

Note: You can repeat this step for all processes that are displayed by the `svcs` command.

```
# svcadm clear svc:/system/ocm:default
```

Setting Up Service State Transition Notifications

To set up the notifications:

1. Ensure that the `smtp-notify` package is installed
2. Enable the notification service
3. Configure the 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`

ORACLE

SMF has a notification feature (also referred to as the notification framework) that notifies you through email messages about service state transitions and fault management events. The notification feature monitors the transition states that are 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 and avoid any possible system down time.

Note: You can also configure the SMF notification feature to work with Simple Network Management Protocol (SNMP) traps. By default, SNMP traps are sent on maintenance transitions. If you use SNMP for monitoring, you can configure additional traps for other state transitions.

To use the notifications feature, you must ensure that the `smtp-notify` package is installed, 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

Verify whether the `system/fault-management/smtp-notify` package is already installed.

```
# pkg info system/fault-management/smtp-notify
```

If the package is not installed, run the following command to install the SMF notification feature:

```
# pkg install system/fault-management/smtp-notify
```


Before you can use the SMF notification feature, ensure that the `smtp-notify` package is already installed on your system. The `smtp-notify` package contains the email notification functionality and the daemon that controls it. To install the package, use the `pkg install` command followed by the package name (`system/fault-management/smtp-notify`), as shown in the example in the slide.

Enabling the `smtp-notify:default` Service

To enable the SMF notification service, run the following command:

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

To confirm whether the service is up and running, run the following command:

```
# ps -ef | grep smtp-notify
    root  3428  1724   0 11:23:37 pts/1   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, use the `svcadm enable` command followed by `svc:/system/fm/smtp-notify:default`, as shown in the example in the slide.

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

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

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

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

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

ORACLE

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 "online" to any other state (`from-online`).

Note: The `svccfg` command is used to modify service configurations. This 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 do not 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 with 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 11 03:34:49 2013
Date: Mon, 11 Nov 2013 03:03:49 +0100 (CET)
From: No Access User
Message-Id: <201211090334.qA93YnCJ001559@s11-server1.mydomain>
Subject: Fault Management Event: solaris:SMF-8000-YX
To: root@solaris.local
Content-Length: 776

SUNW-MSG-ID: SMF-8000-YX, TYPE: defect, VER: 1, SEVERITY: major
EVENT-TIME: Fri Nov 9 03:34:49 UTC 2012
PLATFORM: VirtualBox, CSN: 0, HOSTNAME: s11-server1
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: 473a1ae7-5619-ea1e-dd03-8da51db4fcee
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://support.oracle.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 in this slide) provides information about the event itself. The notification example shown in this slide is alerting you to a change in state for the `apache22` service. As you can see, the notification contains information about the following:

- Severity level of the message (in this example, `major`)
- Time the event took place
- Platform on which the event took place
- Name of the host
- Source
- Event ID
- Description of the event (“`A service failed - a start, stop or refresh method failed.`”)
- The auto-response to the event (“`The service has been placed into the maintenance state.`”)
- Impact of the event (“`svc:/network/http:apache22 is unavailable.`”)
- 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 11 03:34:21 2013
Date: Mon, 11 Nov 2013 03:34:21 GMT
From: No Access User <noaccess@s11-server1.mydomain.com>
Message-Id: <201211090334.qA93YLum001539@s11-server1.mydomain.com>
Subject: s11-server1: svc:/network/http:apache22 online->offline
To: root@s11-server1.mydomain.com
Content-Length:776

HOSTNAME: s11-server1
TIMESTAMP: Mon Nov 11 12:04:23 2013
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 online state to offline state.

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

Managing Service State Transition Notifications

To view the configured notifications, run the following command:

```
# 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 the following command:

```
# 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 in the slide. 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 by using the `delnotify` subcommand, as shown in the second example in the slide. The command shown in the slide stops all notifications.

Quiz

Which of the following commands would you use to determine why a service is in maintenance state?

a. `svcadm`

b. `svccfg`

c. `svcs`

Answer: c

Lesson Agenda

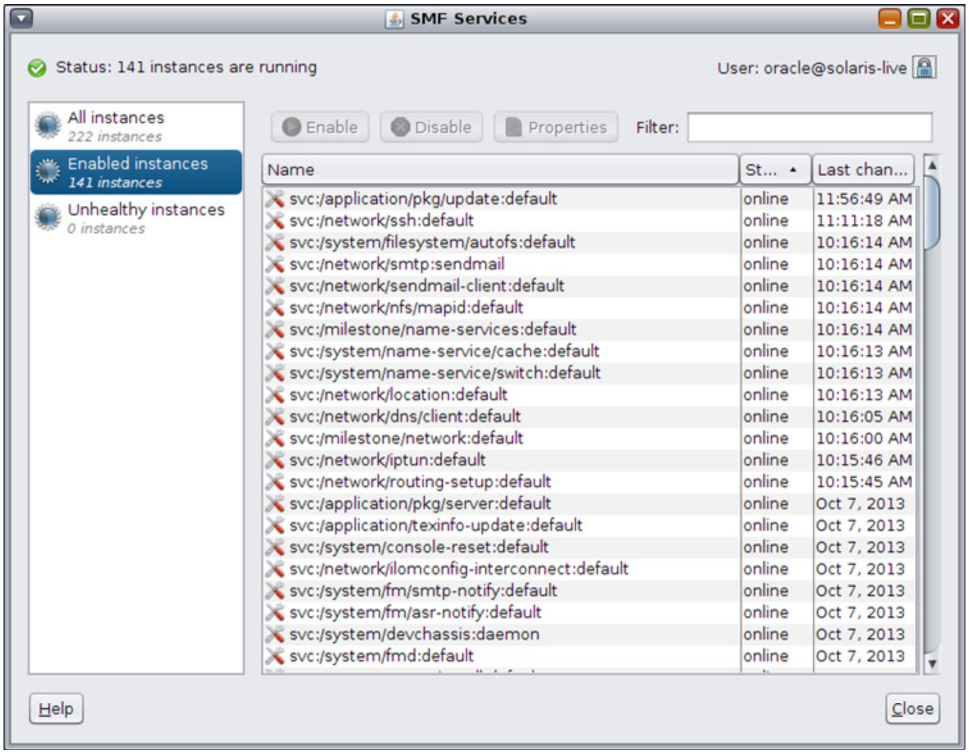
- Describing SMF and Its Components
- Administering SMF Services
- **Managing SMF Services by Using the GUI**

Managing SMF Services by Using the Graphical User Interface

- Introduction to the SMF Graphical User Interface (GUI)
- Managing Service Instances by Using the SMF GUI
- Viewing Service Properties by Using the SMF GUI
- Managing User Credentials by Using the SMF GUI

You can use the SMF GUI to monitor, configure, and perform basic administration of the SMF services or service instances on a local system. The SMF GUI also enables you to connect to a remote system to manage its services.

Introduction to the SMF GUI



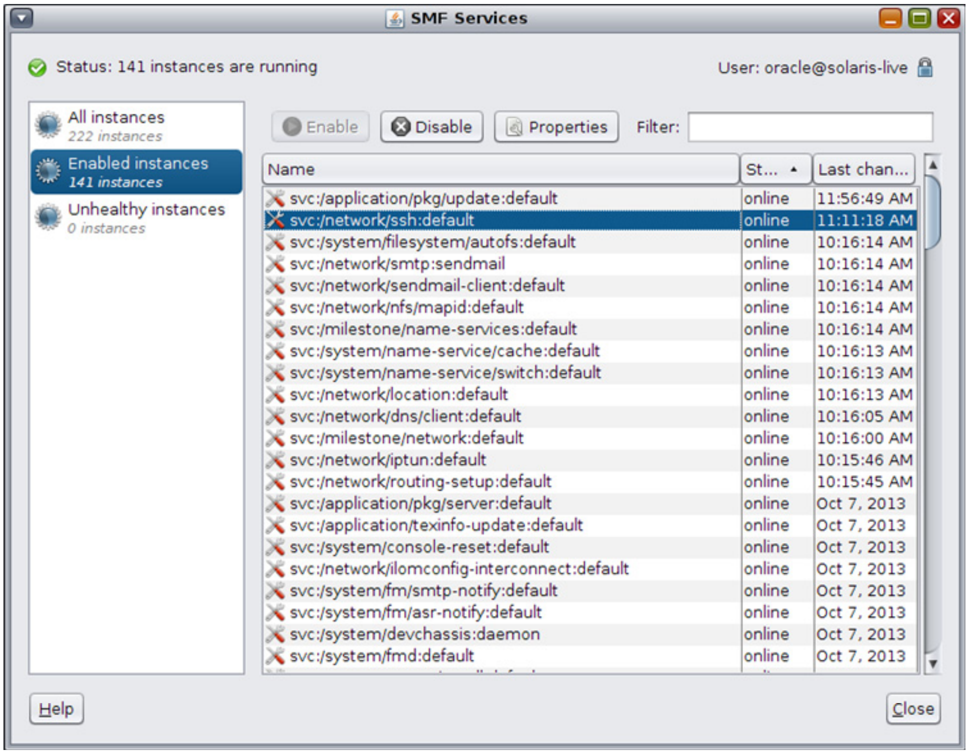
You can access the SMF GUI by using the following menu options from the desktop of Oracle Solaris 11 or later systems:

System > **Administration** > **SMF Service**

The screenshot in the slide shows the default SMF Services window, which is divided into the following three panels:

- **Top panel:** Consists of the following two components:
  - The Status field on the left shows the status of the services running on the local host.
  - The User field on the right shows the credential that is being used by the tool. You can use this field to change the credentials.
- **Left panel:** Enables you to select the group of services that you want to see displayed
- **Right panel:** Displays a list of all services in the group that you select in the left panel

Managing Service Instances by Using the SMF GUI

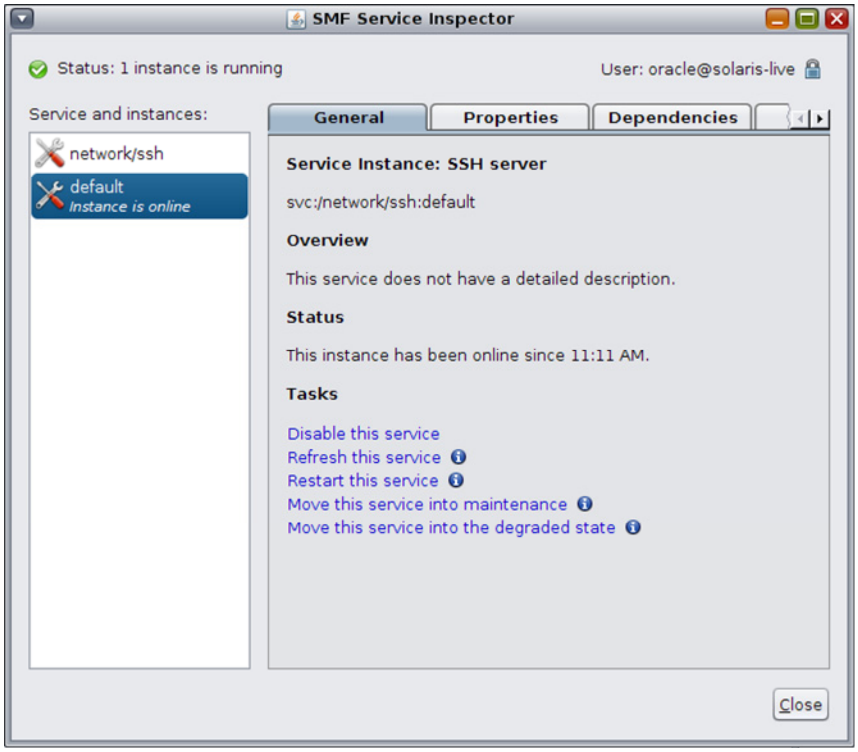


After a group of services is selected in the left panel, a list of service instances is displayed on the right. You can perform the following operations on the service instances:

- Filter services by entering a string of characters in the **Filter** field.
- Sort the list of services by selecting the column header.
- Select a service to enable or disable a service instance.

You can also double-click a service or click the **Properties** button to display additional information about each service or each service instance with its current status and dependencies.

Viewing Service Properties by Using the SMF GUI



ORACLE

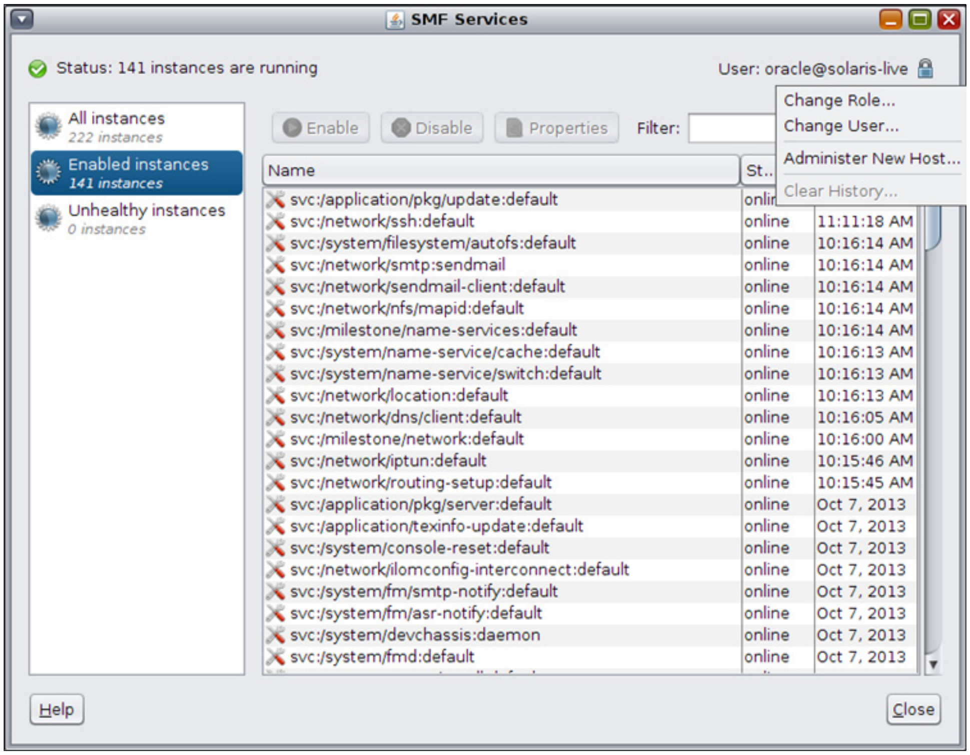
The screenshot in the slide shows generic information about the `default` instance of the `network/ssh` service on the **General** tab.

You can use the **General** tab to perform the following tasks:

- Disable or enable a service.
- Refresh a service.
- Restart a service.
- Move a service into maintenance.
- Move a service into the degrade state.

Additionally, you can access the **Properties**, **Dependencies**, and **Log** tabs to view additional information about the selected service or service instance.

Managing User Credentials by Using the SMF GUI



You can change user credentials by using the **User** field to the right of the top panel in the SMF GUI. Additionally, you can use the **User** field to connect to a remote host and manage its services. The screenshot in the slide shows the user credential options that help you to do the following:

- Change a role.
- Change a user.
- Connect to another host.

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

Practice 4-1: This practice covers the following topics:
- Enabling and disabling services
- Displaying services
- Exploring service dependencies

Practice 4-2: This practice covers the following topics:
- Verifying the installation of required packages
- Configuring the SMF notification
- Examining a service in maintenance

In the practices for this lesson, you are presented with three tasks that are designed to reinforce the concepts presented in the lecture portion. You will have the chance to perform the following tasks:
- **Practice 4-1:** Administering services
- **Practice 4-2:** Administering SMF notifications

You will find Practices 4-1 and 4-2 in your *Activity Guide*. Each practice should take you about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Describe the SMF feature and its components
- Administer SMF services
- Manage SMF services by using the graphical user interface

5

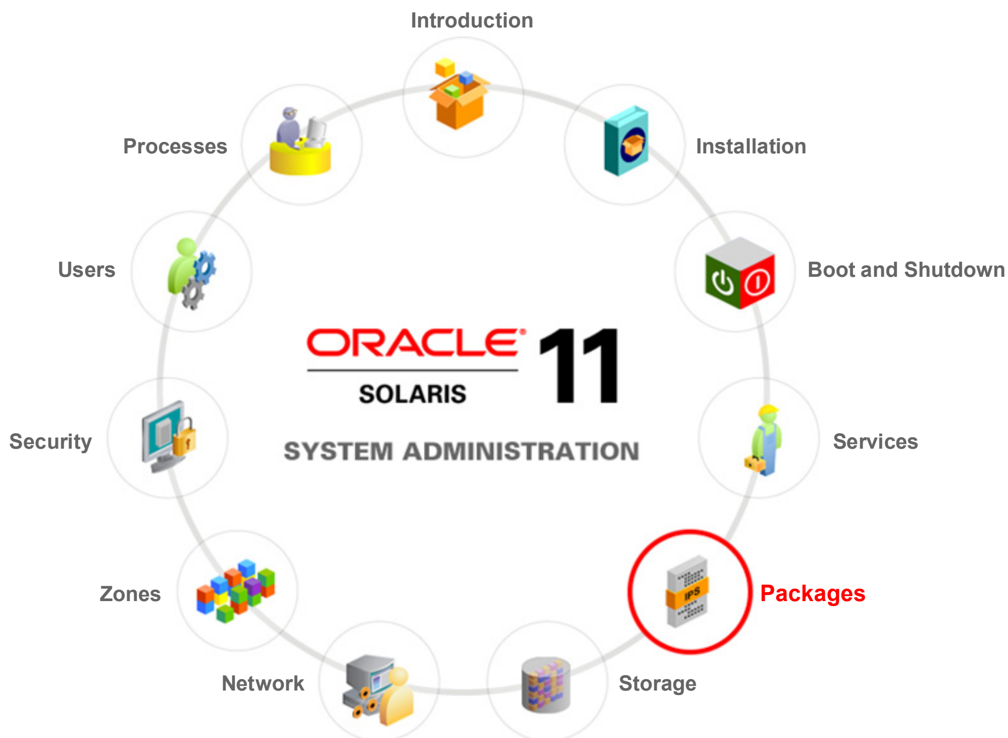
Administering Software Packages by Using IPS

Objectives

After completing this lesson, you should be able to:

- Describe IPS, its components, and interfaces
- Configure an IPS client to access the local IPS repository
- Manage package publishers
- Manage software packages
- Manage signed packages and package properties

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.

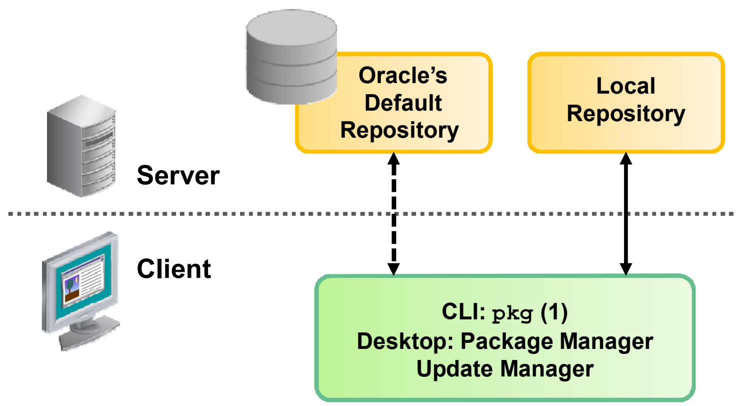
Lesson Agenda

- **Describing IPS, Its Components, and Interfaces**
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- Managing Software Packages
- Managing Signed Packages and Package Properties

Importance of IPS and Package Management

A local IPS repository provides the following benefits:

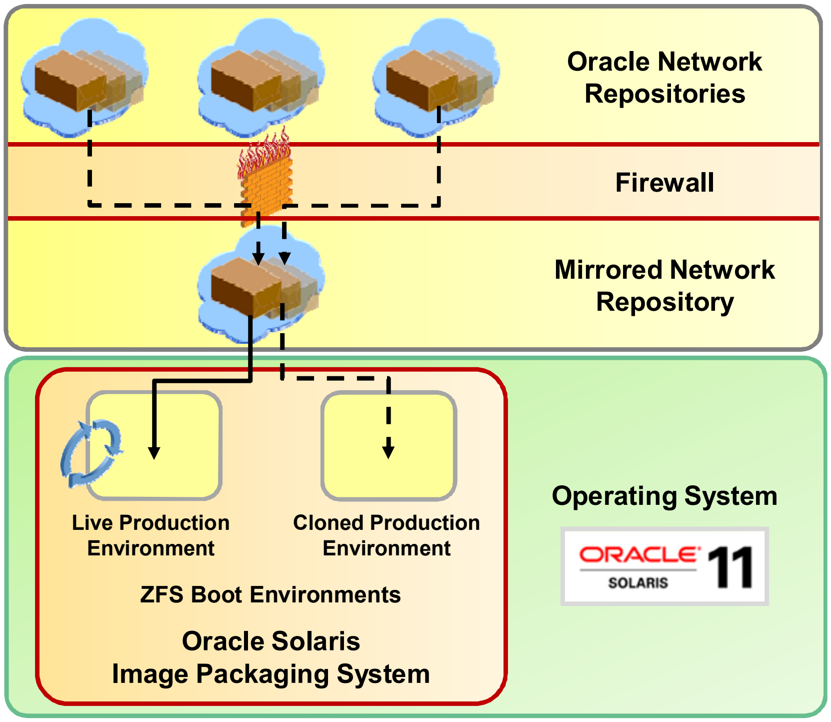
- Performance and security
- Replication
- Customized packages



Every company is concerned about performance and security. A local IPS repository provides the following benefits:

- **Performance and security:** You do not want your client systems to go to the Internet to retrieve new software packages or update existing packages.
- **Replication:** You want to ensure that you can perform the same installation next year that you perform today. You want to easily control the versions to which systems can be updated.
- **Customized packages:** You want to include your own IPS package in the same repository as Oracle Solaris packages.

Introducing IPS



In the earlier releases of the Oracle Solaris platform, administrators used SVR4 packaging to install software onto a system, and then they used a different set of commands to install patches to update the system. As Oracle Solaris evolved to include new technologies, such as Oracle Solaris zones, Oracle Solaris ZFS, and Oracle Solaris Service Management Facility (SMF), previously used processes for managing system updates and upgrades became more complex. With thousands of operating system instances installed in some of today's large virtualized data centers, manual methods of tracking and installing patches could result in errors that negatively affect application availability and security.

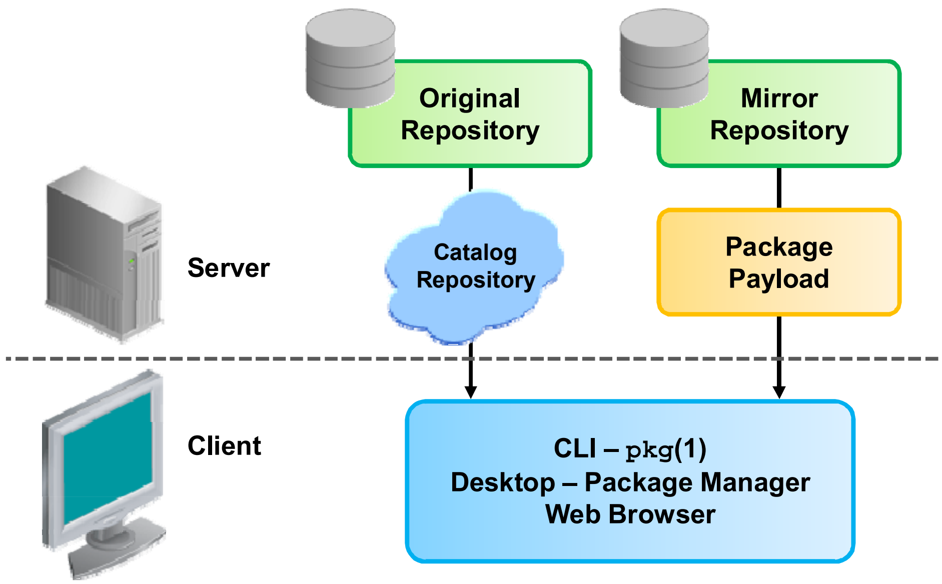
Oracle Solaris 11 takes a new approach to lifecycle and package management to greatly simplify the process of managing system software, thus helping to reduce the risk of operating system maintenance that includes reducing unplanned and planned down time. With Image Packaging System (IPS), administrators can install and update software from locally connected or remote software package repositories by using a much-improved and modernized process.

The Image Packaging System (IPS) is a network-based package management system that is included in Oracle Solaris 11. It provides a framework for complete software lifecycle management such as installation, upgrade, and removal of software packages. IPS is an integrated solution that helps automate and ease the complexity of managing system software on Oracle Solaris 11 by integrating patching with package updates. It provides a network-centric and efficient approach to automatic software dependency checking and validation, and it builds on the foundation of ZFS as the default root file system. Using IPS, you can easily and reliably install or replicate an exact set of software package versions across many different client machines, and get a much clearer understanding of any differences between the software versions installed on a system.

With IPS, you can perform the following tasks:

- Create and manage images.
  Note: An image is a bootable instance of the Oracle Solaris 11 operating system.
- Search the IPS packages on your system and in IPS repositories.
- Copy, mirror, create, and administer package repositories.
- Create and publish IPS packages to a package repository.
- Republish the content of an existing package in a package repository.
- Upgrade a system with the help of ZFS boot environments.

Introducing IPS Components



IPS is made up of the following key components that have important roles to play in package management:

- **Manifest:** A manifest is a text file that describes the components and attributes that make up a package.
- **Package:** An IPS package is defined by a manifest. A package manifest describes package actions in a defined format of key/value pairs and possibly a data payload. Package actions include files, directories, links, drivers, dependencies, groups, users, and license information. Package actions represent the installable objects of a package. Actions called `set` actions define package metadata such as classification, summary, and description. The types of packages are as follows:
  - **Incorporation packages:** An incorporation package specifies the versions of the other packages that can be installed. Installing an incorporation package does not install any other packages. An incorporation package ensures that if you install an incorporate dependency package of that incorporation package, only the prescribed version of the dependent package can be installed. For example, if a package specified as an incorporate dependency in an installed incorporation package is version 1.4.3, no version less than 1.4.3 or greater than or equal to 1.4.4 can be installed. Versions that merely extend the dotted sequence, such as 1.4.3.7, could be installed.

- **Group packages:** A group package specifies the set of packages that constitute a feature or tool. Installing a group package installs all the group dependency packages in that group package. Packages that are specified as group dependencies in a group package do not specify the package version. The group package is a content management tool, not a version management tool.

- **Repository:** A repository is a location where clients publish packages to and retrieve packages from. The location is described by a uniform resource identifier (URI) such as http://pkg.oracle.com/solaris/release. A repository is also called a *depot server*. It contains packages from a single publisher (for example, Solaris). A publisher can publish to multiple repositories. A repository has an origin and zero or more mirrors. The repository origin is the location of a package repository that contains both package metadata (package manifests and catalogs) and package content (package files). A mirror is a location of a package repository that contains only package content. For information about creating and publishing a package to the IPS repository, refer to http://www.oracle.com/technetwork/articles/servers-storage-dev/int-s11-repositories-1632678.html.
- **FMRI:** Each IPS package is represented by a Fault Management Resource Identifier (FMRI). FMRIs are used with the `pkg`(1) command to indicate which packages to perform operations on. 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@0.5.11,5.11-0.175.2.0.0.24.1:20131001T50456Z`, 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:** `0.175.2.0.0.24.1`
  - **Time Stamp** (when the package was published)**:** `20131001T50456Z`

  For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E36802/fmri.html#scrolltoc.
- **Catalog:** A catalog consists of all packages in an IPS package repository. The packages in a catalog are associated with a specific publisher.
- **Mirror:** A mirror provides a subset of the data that origins provide. Mirrors can be used only for downloading package files. Package metadata is downloaded from the origin. IPS clients access the origin to obtain a publisher's catalog, even when the clients download package content from a mirror.

- **Client package management utilities:**
  - **`pkg`(1):** This is a command-line command that can be used to create and manage images, search package data, and perform software installation, upgrade, and removal.
  - **Package Manager:** The Package Manager application provides a graphical user interface (GUI) for IPS. It also provides a subset of the functionality offered by the command-line commands provided with IPS.
  - **Web browser:** A web browser can be used to search for and install software packages from an IPS repository.

Introducing the IPS Interfaces

IPS supports the following interfaces:

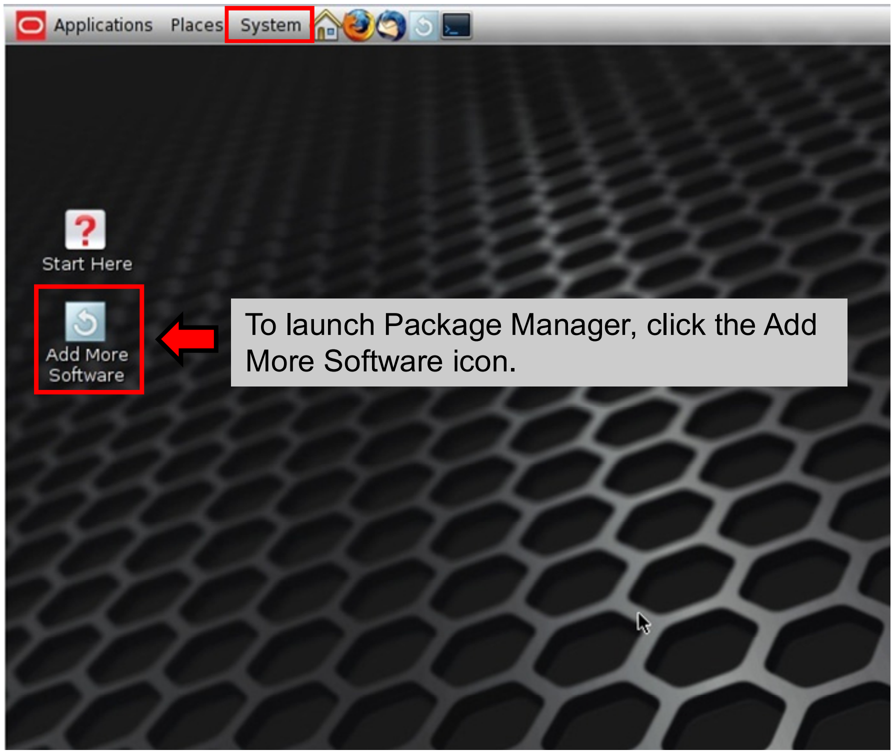
- Command line
- GUI
  - Package Manager
  - Update Manager
- Brower User Interface (BUI)

With IPS, you have the option of using a command-line interface (CLI), graphical user interface (GUI), or browser user interface (BUI) to perform package management tasks. IPS supports two GUIs:

- Package Manager
- Update Manager

You will learn how to use the CLI commands and GUI later in this lesson and you will have the opportunity to work with them in the practices.

Package Manager



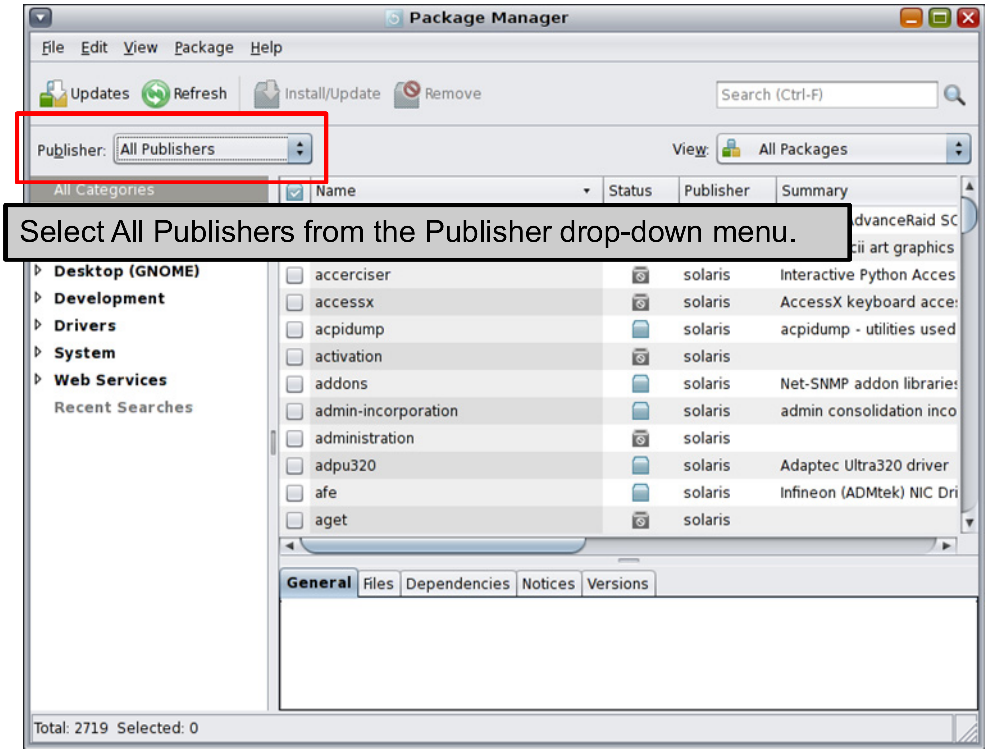
From a desktop environment, you can invoke Package Manager either directly from the System menu on the toolbar (System > Administration > Package Manager) or from the Package Manager GUI. To access Package Manager, click the Add More Software icon.

Note:

You can also access Package Manager from the command line by using the `/usr/lib/pm-launch /usr/bin/packagemanager` command. For more information about the Package Manager command-line options, refer to http://docs.oracle.com/cd/E36784_01/html/E36802/learn_ips.html#AUOSSpkg_mgr.

The `packagemanager`(1) command and its associated command `pm-updatemanager`(1) might be removed in a future Oracle Solaris release. Instead, you can use `pkg`(1), which provides access to all package management capabilities.

Package Manager



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

Update Manager

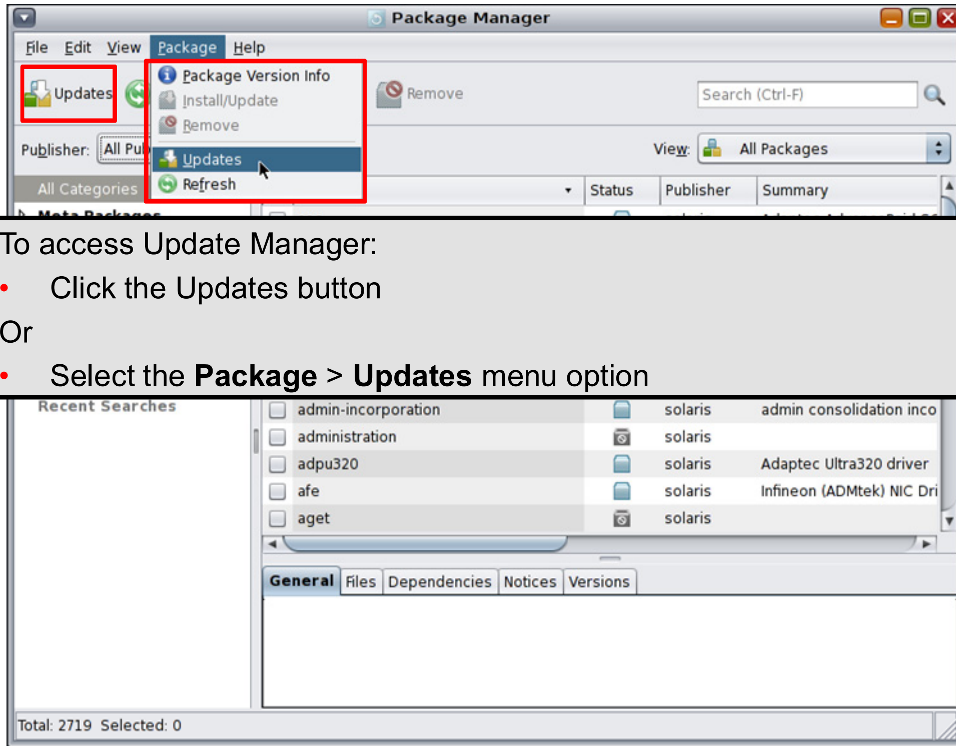
- Is used to update all installed packages to the latest version
- Can be invoked in several ways:
  - CLI command
  - Package Manager GUI

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 the Package Manager GUI.

To access Update Manager from the command line, use the `/usr/lib/pm-launch /usr/bin/pm-updatemanager` command. For more information about the Update Manager command-line options, refer to http://docs.oracle.com/cd/E36784_01/html/E36802/learn_ips.html#AUOSSpkg_mgr.

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

Update Manager



To access Update Manager:

- Click the Updates button

Or

- Select the **Package** > **Updates** menu option

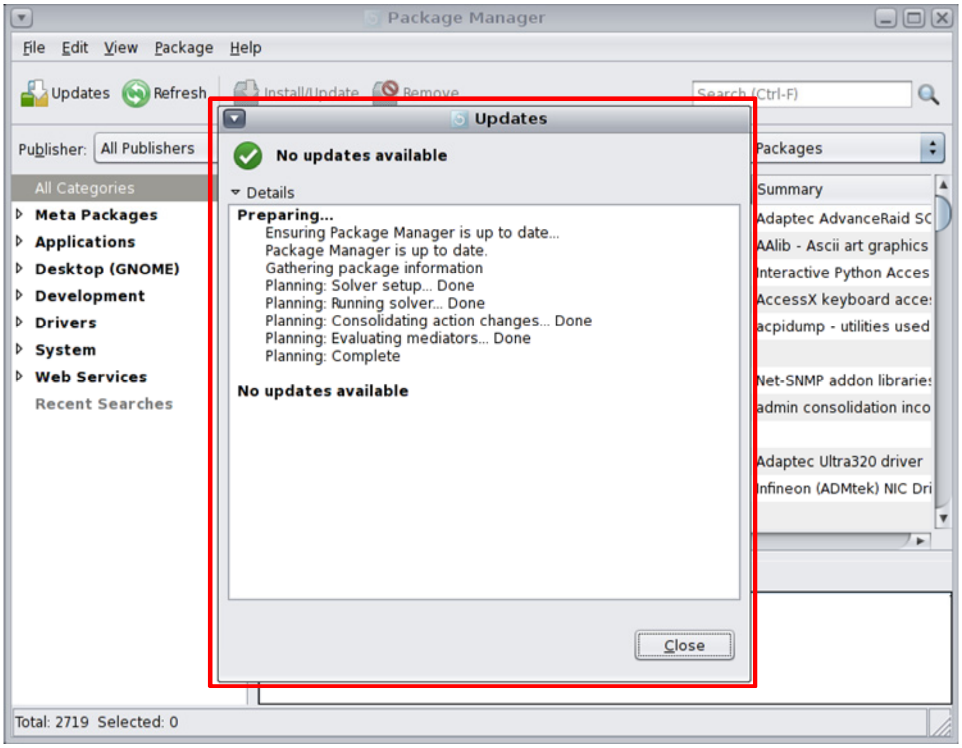
ORACLE

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).

Update Manager

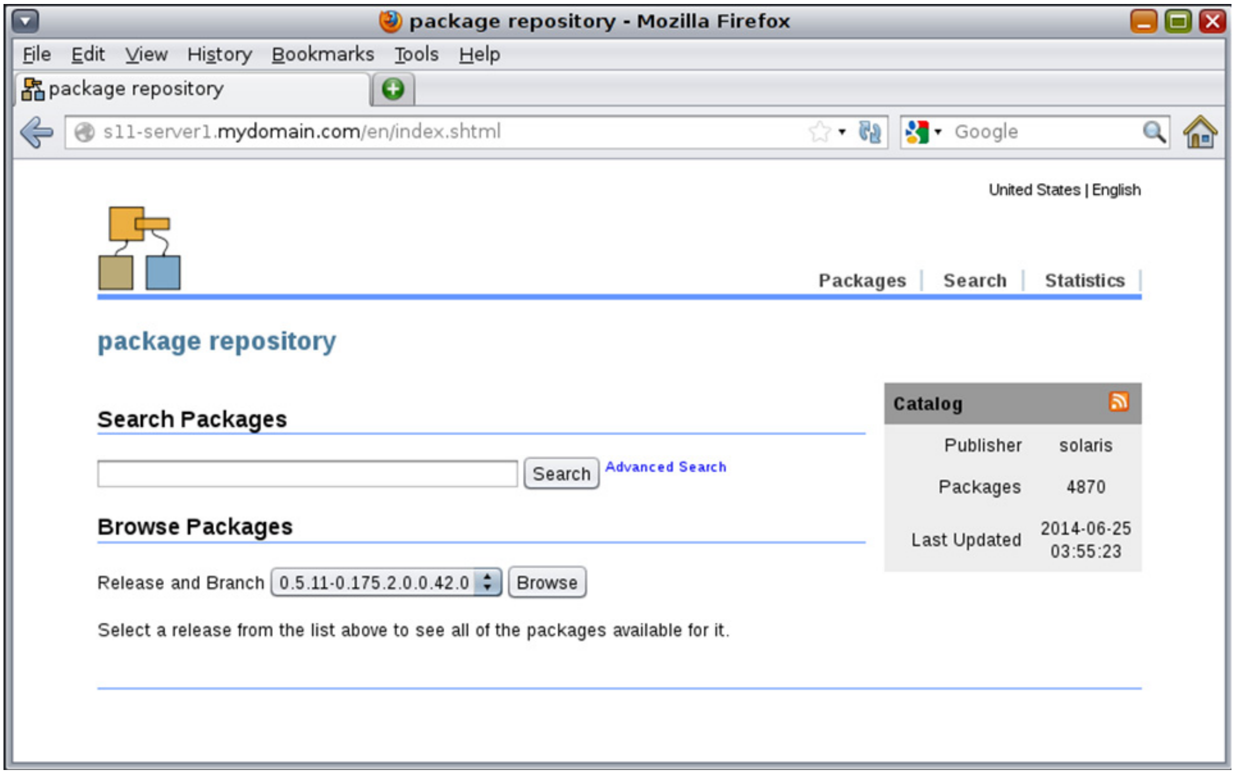


IPS runs through a series of checks before it actually starts the update process. The first thing IPS does is to make sure that Package Manager is up-to-date by accessing the catalogs for each of the publishers, which will identify any available updates. The packages in a catalog are associated with 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`.

With the help of Update Manager, you can also update the operating system, which is covered in depth in the *Oracle Solaris 11 Advanced System Administration* course, which is a follow-on to this course.

Accessing the Package Repository with a BUI



IPS allows you to access the package repository by using a BUI. With a BUI, you can search for and install packages, and view the contents of a package manifest.

Lesson Agenda

- Describing IPS, Its Components, and Interfaces
- **Configuring an IPS Client to Access the Local IPS Repository**
- Managing Package Publishers
- Managing Software Packages
- Managing Signed Packages and Package Properties

Configuring an IPS Client to Access the Local IPS Repository

Required tasks:

1. Determining the client host and domain names
2. Checking network connectivity
3. Setting the publisher
4. Testing client access to the local IPS server

Note: You learn about configuring a local IPS repository in the follow-on course *Oracle Solaris 11 Advanced System Administration.*

Determining the Client Host and Domain Names

Use the `hostname` and `domainname` commands to identify the client machine.

```
# hostname
s11-desktop

# domainname
mydomain.com
```


Your first task is to identify the client machine's host and domain names. To do this, run the `hostname` and `domainname` commands, as shown in the slide.

Checking Network Connectivity

Verify DNS service access and connectivity with the local IPS server.

```
# nslookup s11-server1
Server:         192.168.0.100
Address:        192.168.0.100#53

Name:           s11-server1.mydomain.com
Address:        192.168.0.100

# ping s11-server1
s11-server1 is alive
```


Next, you want to verify that the client machine can access DNS services and that it can connect to the local IPS server. For information about how to set up the DNS server, refer to http://docs.oracle.com/cd/E36784_01/html/E36831/dnsref-31.html#scrolltoc.

First, run the `nslookup` command with the name of the local IPS server. In the slide example, the local IPS server host name is `server`.

Then verify that the client can talk with the local IPS server by running the `ping` command, as shown in the example.

Setting the Publisher

Use the `pkg set-publisher` command to set the publisher to point to the local IPS repository.

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

# pkg set-publisher -G '*' -g http://s11-server1.mydomain.com/ solaris

# pkg publisher
PUBLISHER     TYPE         STATUS P LOCATION
solaris       origin       online F http://s11-server1.mydomain.com/
```

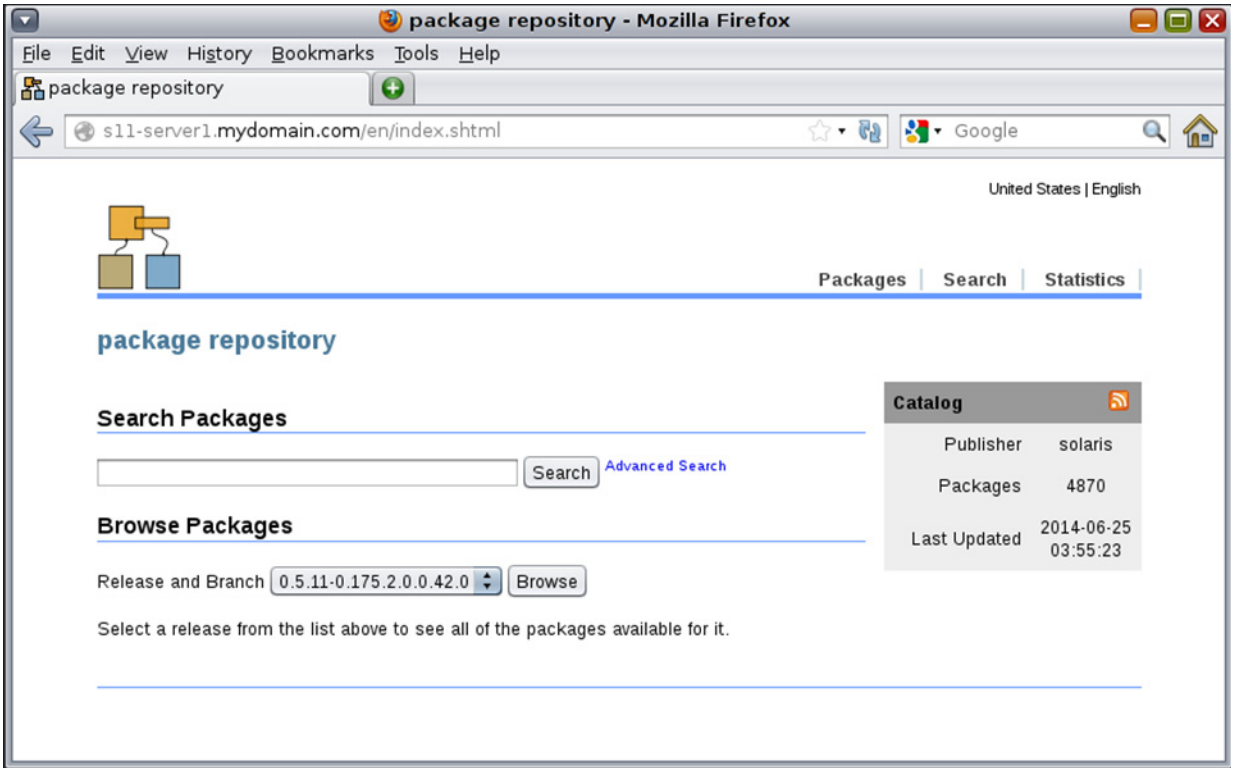

After you have verified network connectivity between the client and the local IPS server, you need to set the publisher to the local IPS publisher, just as you did when you configured the local IPS server.

First, check the current publisher. Next, set the publisher to the local IPS repository, and then verify that the publisher is now the local IPS publisher.

http://pkg.oracle.com/solaris/release is the default online package repository that is provided by Oracle. If you have a support contract, you can connect to http://pkg.oracle.com/solaris/support. You learn more about these repositories and how you use them to update your system OS in the *Oracle Solaris 11 Advanced System Administration* course.

Make a note of the local publisher's URI; you will need it to complete the next task.

Testing Client Access to the Local IPS Server



ORACLE

The final task is to test the client machine's access to the local IPS server. To do this, open the local publisher's URI in a browser. If a page is returned that reads "package repository," you have successfully configured the client's access to the IPS server. You can now use your local IPS repository to manage your company's software package needs.

Note: In this example, the URI would be http://s11-server1.mydomain.com.

Practice 5-1 Overview: Configuring an IPS Client to Access the Local IPS Server

This practice covers the following topics:

- Verifying connectivity between the client and the IPS server
- Removing and adding publishers
- Testing client access to the IPS server
- Searching for packages by using the package repository browser

This practice should take about 30 minutes to complete.

Lesson Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- **Managing Package Publishers**
- Managing Software Packages
- Managing Signed Packages and Package Properties

Managing Package Publishers

This section covers the following topics:

- Displaying publisher information
- Specifying publisher rankings
- Specifying publisher stickiness
- Setting the publisher search order
- Disabling or enabling a publisher
- Changing a publisher's origin URI

Displaying Publisher Information

- To display only the highest-ranked publisher in the search order, run `pkg publisher -P`.

```
# pkg publisher -P
PUBLISHER              TYPE     STATUS   P LOCATION
solaris                origin   online   F http://s11-server1.mydomain.com/
```

- To display information about a specific publisher, run `pkg publisher` *`publisher_name`*.

```
# pkg publisher solaris
       Publisher: solaris
           Alias:
      Origin URI: http://s11-server1.mydomain.com/
         SSL Key: None
        SSL Cert: None
     Client UUID: 55dc8a86-fbe5-11e3-bc11-811d2d030777
 Catalog Updated: June 25, 2014 03:55:23 AM
         Enabled: Yes
```


If you are interested in seeing information only about the highest-ranked publisher in the search order, you can run the `pkg publisher` command with the uppercase `-P` option, as shown in the first example in the slide. You can see the publisher name, type, status, and URI.

If you have multiple publishers and you want to see information about a particular publisher, you can use the `pkg publisher` command with the publisher name. As shown in the second example in the slide, this command displays the publisher's name, an alias if one has been assigned, the origin URI, the publisher's SSL keys and certificates information if there are any, the client's universally unique identifier (UUID), the date the publisher's catalog was last updated, and whether the publisher is enabled.

Note: For information about specifying SSL keys and certificates for a publisher, see the Image Packaging System man pages.

Specifying Publisher Rankings

To set a publisher to be the highest-ranked publisher in the search order, run `pkg set-publisher -P` *`publisher_name`* or the `--search-first` option.

```
# pkg publisher
PUBLISHER                TYPE    STATUS   P LOCATION
solaris                  origin  online   F http://s11-server1.mydomain.com
whoisit.com (non-sticky) origin  online   F http://pkg.example.com/release

# pkg set-publisher -P whoisit.com

# pkg publisher
PUBLISHER                TYPE    STATUS   P LOCATION
whoisit.com              origin  online   F http://pkg.example.com/release
Solaris (non-sticky)     origin  online   F http://s11-server1.mydomain.com
```


As you may recall, when you first looked at publisher information while configuring the local IPS server, the publisher was the default `solaris` publisher from Oracle. You then removed the default publisher origin for the `solaris` publisher and added a new origin to it. The `solaris` publisher then became the highest-ranked publisher in the search order (primarily because it was the only publisher at that point).

Assume that you have since added several publishers to your local IPS server and you want to specify a different publisher as the highest-ranked publisher. To make this change, you use the `-P` option with the `pkg set-publisher` command, followed by the publisher's name, as shown in the example in the slide, where you make the `whoisit.com` publisher the highest-ranked publisher. Note how the specified publisher moves to the top of the search order when you run the `pkg publisher` command again. (The "non-sticky" specification is discussed on the next page.)

Note: You can specify the `-P` option when you add a publisher or when you modify an existing publisher. Also, you can choose to use the `--search-first` option to search for the higher-ranked publisher.

Specifying Publisher Stickiness

To make a publisher sticky, run `pkg set-publisher --sticky` *`publisher_name`*.

```
# pkg set-publisher --sticky example.com
```

To make a publisher non-sticky, run `pkg set-publisher --non-sticky` *`publisher_name`*.

```
# pkg set-publisher --non-sticky example.com
```

ORACLE

To make sure that a package that was installed from one publisher cannot be updated by another publisher, you can make the publisher "sticky." To do this, you use the `pkg set-publisher --sticky` command, followed by the publisher's name (as shown in the first example in the slide). A newly added publisher is sticky by default.

Conversely, if you want to allow a package that was installed from one publisher to be updated by another publisher (so that publishers that are ranked higher than the specified publisher can provide updates to packages originally installed from the specified publisher), you can use the `pkg set-publisher` command with the `--non-sticky` option (as shown in the second example).

Setting the Publisher Search Order

To move a publisher higher in the search order, run `pkg set-publisher --search-before` *`publisher_name publisher_name`*.

```
# pkg set-publisher --search-before example1.com example2.com
```

To move a publisher lower in the search order, run `pkg set-publisher --search-after` *`publisher_name publisher_name`*.

```
# pkg set-publisher --search-after example1.com example2.com
```


The publisher search order is used to find packages to install. The publisher search order is also used to find packages to update if the publisher that the package was originally installed from is non-sticky.

The high ranked publisher is first in the search order. A newly added publisher is the last in the search order by default.

To move a publisher higher in the search order, use the `pkg set-publisher --search-before` command, followed by the name of the publisher that you want to be demoted in the search order, and then the name of the publisher that you want to be promoted in the search order (as shown in the first example in the slide).

To move a publisher lower in the search order, use the `pkg set-publisher --search-after` command, followed by the name of the publisher that you want to be searched first, and then the name of the publisher that you want to be searched thereafter (as shown in the second example).

Disabling and Enabling a Publisher

To disable a publisher, run `pkg set-publisher -d` *`publisher_name`*.

```
# pkg set-publisher -d solaris.com
```

To enable a publisher, run `pkg set-publisher -e` *`publisher_name`*.

```
# pkg set-publisher -e solaris.com
```

ORACLE

There may be times when you need to disable a publisher temporarily. Suppose that you have a planned server or network outage. To minimize client down time to other publishers, you decide to temporarily disable your publisher. After the planned down-time window is past, you can enable the publisher again.

To disable a publisher, use the `pkg set-publisher` command with the `-d` option, followed by the publisher's name, as shown in the first example in the slide. A disabled publisher is not used in package operations, such as `list` and `install`. However, you can still modify the properties of a disabled publisher.

Note: The highest-ranked publisher cannot be disabled.

Use the `-e` option with the `pkg set-publisher` command to enable a publisher, as shown in the second example.

Note: To see the disabled or enabled status of a publisher, you can run the `pkg publisher` command with the publisher's name.

Changing a Publisher's Origin URI

To change a publisher's origin URI, run `pkg set-publisher -g` *`newpublisher_URI`* `-G` *`oldpublisher_URI`* *`newpublisher`*.

```
# pkg set-publisher -g http://pkg.example.com/support \
-G http://pkg.example.com/release solaris
```


There may be times when you want or need to change a publisher's origin URI. For example, you want to switch `pkg.example.com/release` to `pkg.example.com/support`. The command to change a publisher's origin URI is very similar to the command that you used to add a new publisher to the local IPS server and remove the default `solaris` publisher. To change the origin URI for a publisher, you add the new URI by using the `-g` option and remove the old URI by using the `-G` option, as shown in the example in the slide.

Quiz

You want to set `mypublisher.com` as the highest-ranked publisher for your local IPS repository. Which command would you use to execute this task?

a. `pkg publisher -P mypublisher.com`

b. `pkg publisher -n mypublisher.com`

c. `pkg set-publisher -P mypublisher.com`

d. `pkg set-publisher -e mypublisher.com`

Answer: c

Quiz

You have three publishers listed in the following order: `mypublisher.com` (the highest-ranked publisher), `solaris`, and `whoisit`. For search order purposes, you want to move the `whoisit` publisher before the `solaris` publisher. Which command would you use to execute this task?

a. `pkg set-publisher --search-before solaris whoisit`

b. `pkg set-publisher --search-after solaris whoisit`

c. `pkg set-publisher --search-before whoisit solaris`

Answer: a

Lesson Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- **Managing Software Packages**
- Managing Signed Packages and Package Properties

Managing Software Packages by Using the CLI

- Listing package state information
- Displaying package information
- Displaying contents of a package
- Updating and installing 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

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

```
# pkg list entire
NAME (PUBLISHER)      VERSION                        IFO
entire                0.5.11-0.175.2.0.0.42.0        i--
```


To list package state information, use the `pkg list` command with the Fault Management Resource Identifier (FMRI) of the package, 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@0.5.11,5.11-0.175.2.0.0.24.1:20131001T50456Z`, 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:** `0.175.2.0.0.24.1`
- **Time Stamp** (when the package was published)**:** `20131001T50456Z`

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

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: Not installed
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
        Branch: 0.175.2.0.0.42.2
Packaging Date: June 24, 2014 06:39:19 PM
          Size: 162.04 kB
          FMRI: pkg://solaris/developer/apptrace@0.5.11,5.11-
    0.175.2.0.0.42.2:20140624T183919Z
```


To display information about a package, use the `pkg info` command with the FMRI of the package. In the example in the slide, 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 that are installed on the system, use `pkg info` without a package FMRI.

Displaying the Contents of a Package

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

```
# pkg contents compress/zip
PATH
usr
usr/bin
usr/bin/zip
usr/bin/zipcloak
usr/bin/zipnote
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 in the slide, 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

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

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


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

Note: 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)

Viewing an Installation Action Without Installing

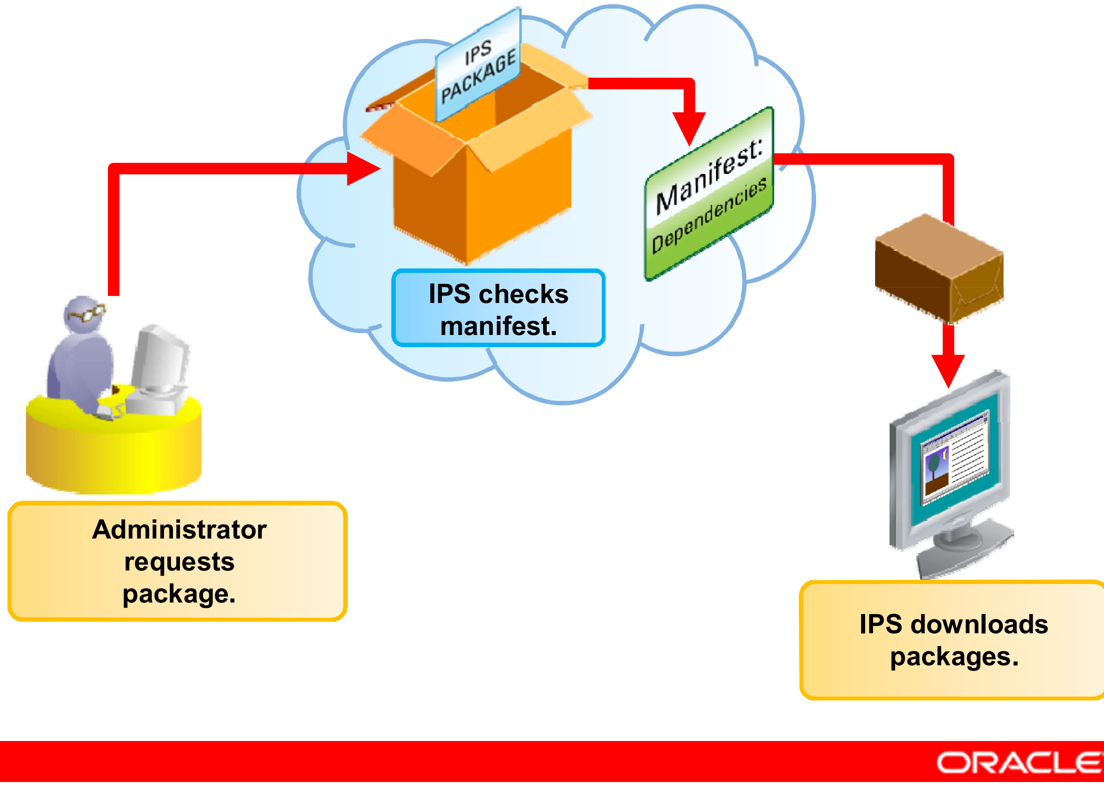
To view an installation action without installing the package, run `pkg install -n` *`pkg-fmri`*.

```
# pkg install -nv apptrace
           Packages to install:           1
      Estimated space available:    31.96 GB
Estimated space to be consumed:     19.83 MB
        Create boot environment:          No
Create backup boot environment:          No
           Rebuild boot archive:          No

Changed packages:
solaris
   developer/apptrace
     None -> 0.5.11,5.11-0.175.2.0.0.42.2:20140624T183919Z
```


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 in the slide.

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

Installing 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 that 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.

Installing a Package

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)   SPEED
Completed                     1/1       10/10    0.1/0.1    31.6k/s

PHASE                                          ITEMS
Install new actions                            29/29
Updating package state database                 Done
Updating package cache                           0/0
Updating image state                            Done
Creating fast lookup database                   Done
Updating package cache                           1/1
```

ORACLE

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

With a package update, by default, the latest 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)

Verifying a Package Installation

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

```
# pkg verify -v apptrace
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.

Notes

- If you want to run a `verify` against multiple packages, you can specify more than one *`pkg-fmri`* pattern. If you want to display verbose information messages, you can use the `-v` option, and to display only error messages, you can use the `-q` option.
- Use the `pkg fix` command to fix the package errors that are reported by the `pkg verify` command. For more information, refer to the `pkg`(1) man page and http://docs.oracle.com/cd/E36784_01/html/E36802/gilmv.html#scrolltoc.

Searching for a Package

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

```
# pkg search -l bash
INDEX                  ACTION VALUE              PACKAGE
basename               dir    etc/bash           pkg:/shell/bash@4.1.11-0.175.2.0.0.42.1
basename               dir    usr/share/bash     pkg:/shell/bash@4.1.11-0.175.2.0.0.42.1
basename               file   usr/bin/bash       pkg:/shell/bash@4.1.11-0.175.2.0.0.42.1
com.oracle.info.name   set    bash               pkg:/shell/bash@4.1.11-0.175.2.0.0.42.1
pkg.fmri               set    solaris/shell/bash pkg:/shell/bash@4.1.11-0.175.2.0.0.42.1
```


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 in which 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

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
Removing old actions                            25/25
Updating package state database                  Done
Updating package cache                            1/1
Updating image state                             Done
Creating fast lookup database                    Done
Updating package cache                            1/1
```


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

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

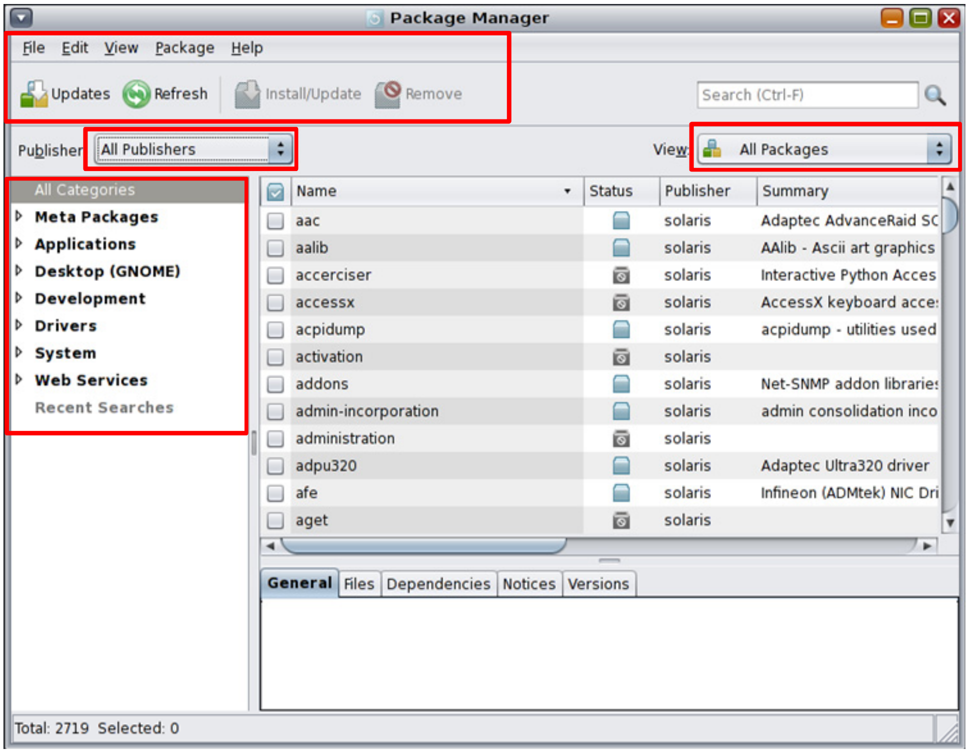
Package Management Commands: Summary

Package Management Task	IPS Command
Display package state and version information.	`pkg list`
Display package information.	`pkg info`
Display the contents of a package.	`pkg contents`
Install package updates.	`pkg update`
Install the package.	`pkg install`
Verify the 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 that are presented in this lesson. You now learn how to perform these tasks by using Package Manager.

Note: Remember that although Package Manager supports many of these same package management tasks, it organizes and displays package information in slightly different ways.

Managing Packages by Using the Package Manager GUI



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. On 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:

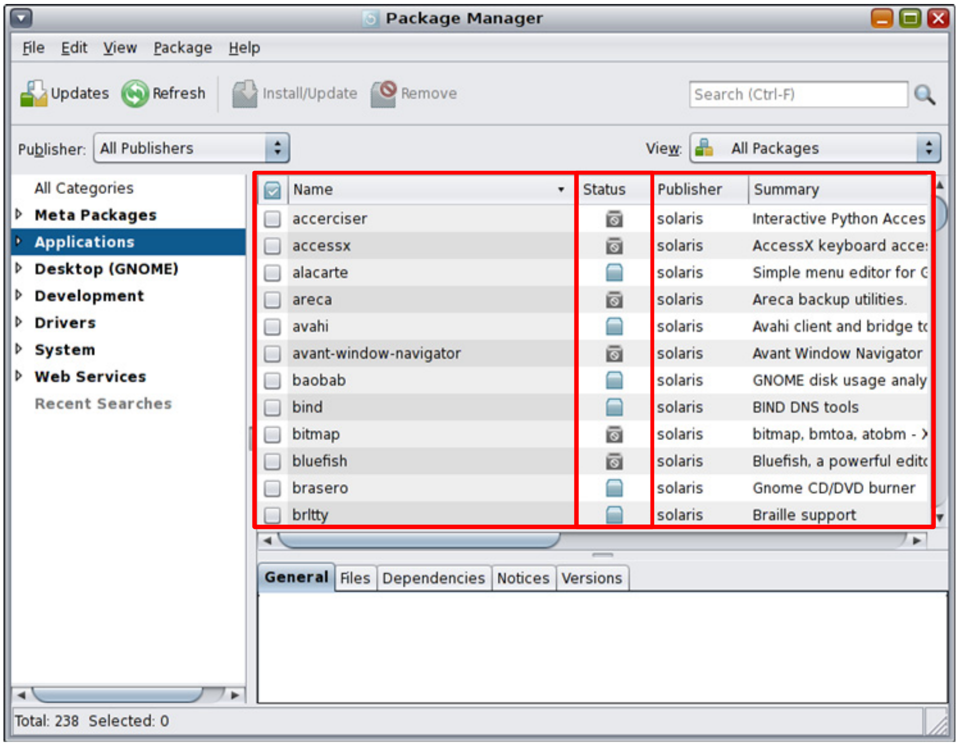
- **Updates:** Enables you to retrieve package updates
- **Refresh:** Enables you to refresh the GUI
- **Install/Update:** Enables you to perform the install/update function on a selected package or packages
- **Remove:** Uninstalls a selected package or packages

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 extreme right of the Publisher button enables you to view all packages, installed packages only, updates only, uninstalled packages, and packages that 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 only application packages, desktop packages, development packages, and so on. Within each category, there are subcategories that enable you to restrict your list of packages further.

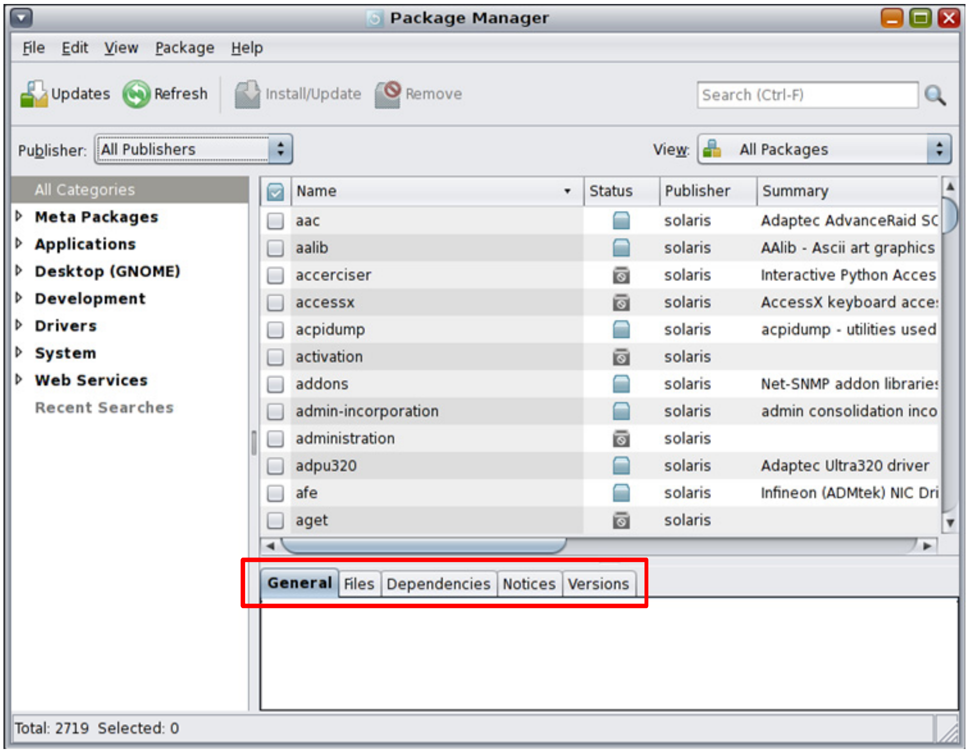
Managing Packages by Using the Package Manager GUI



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 slide screenshot.

The name, status, publisher, 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 is not installed, an “inactive” folder icon is displayed; the folder is disabled and has a “not installed” symbol displayed on it.

Managing Packages by Using the Package Manager GUI

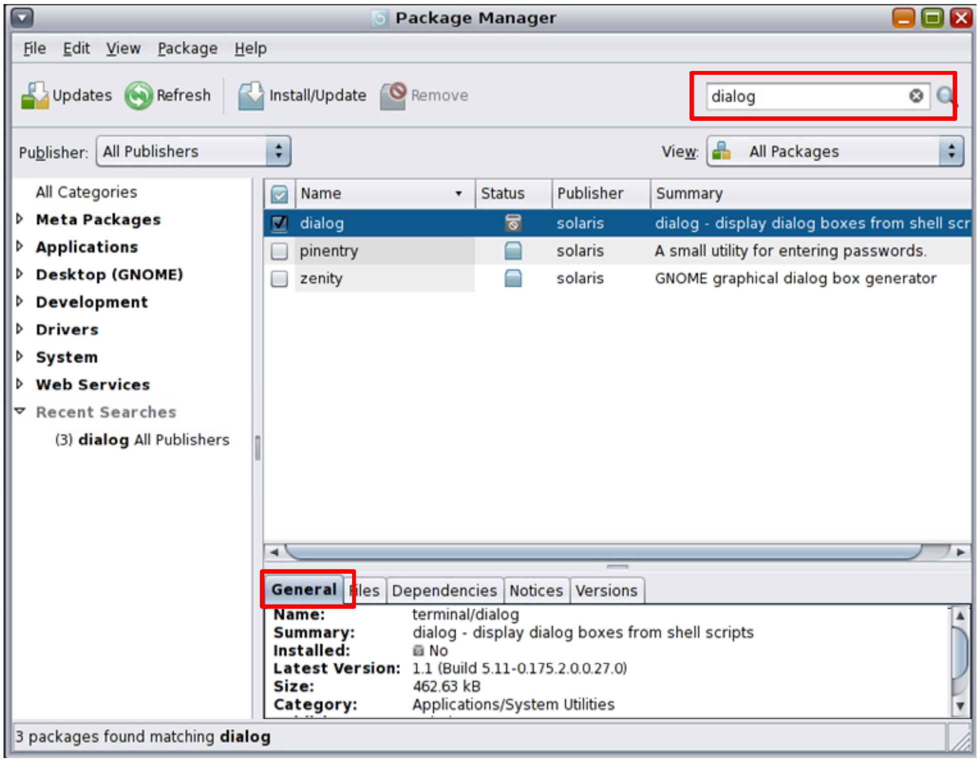


To see information that is specific to a file, you can use the tabs that appear below the list of packages. The tabs from left to right are:

- **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



To search for the “dialog” package, enter a keyword in the search field and click the Browse icon (magnifying glass). If it is found, the package is displayed in the main window.

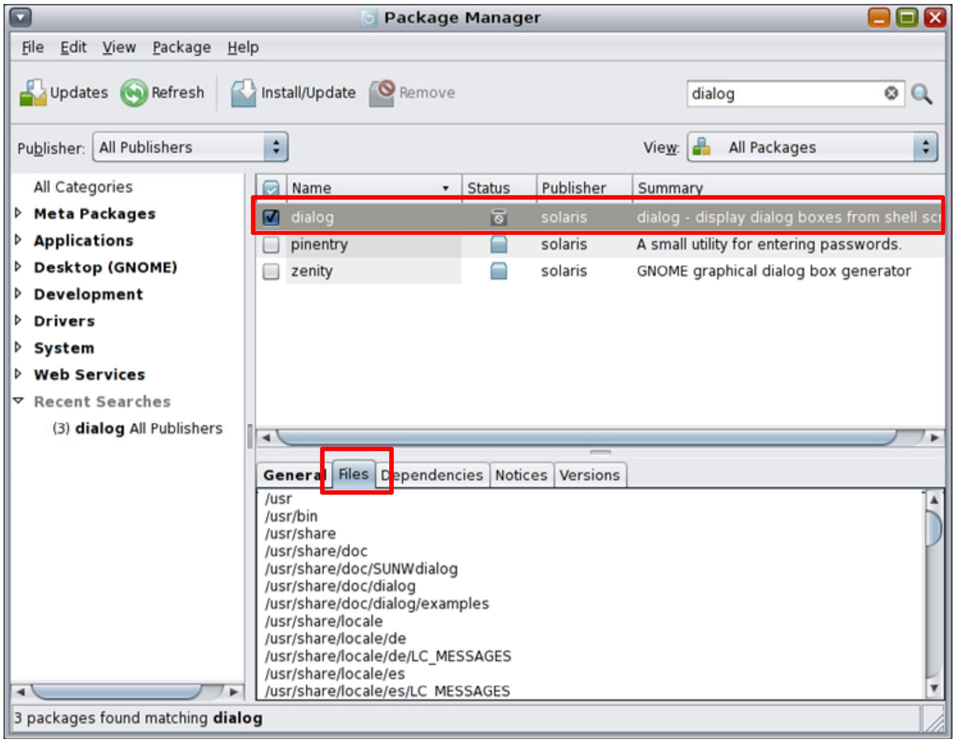
To display information about the package, select the package and click the General tab, as shown in the example in the slide.

The following information is displayed on the General tab:

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

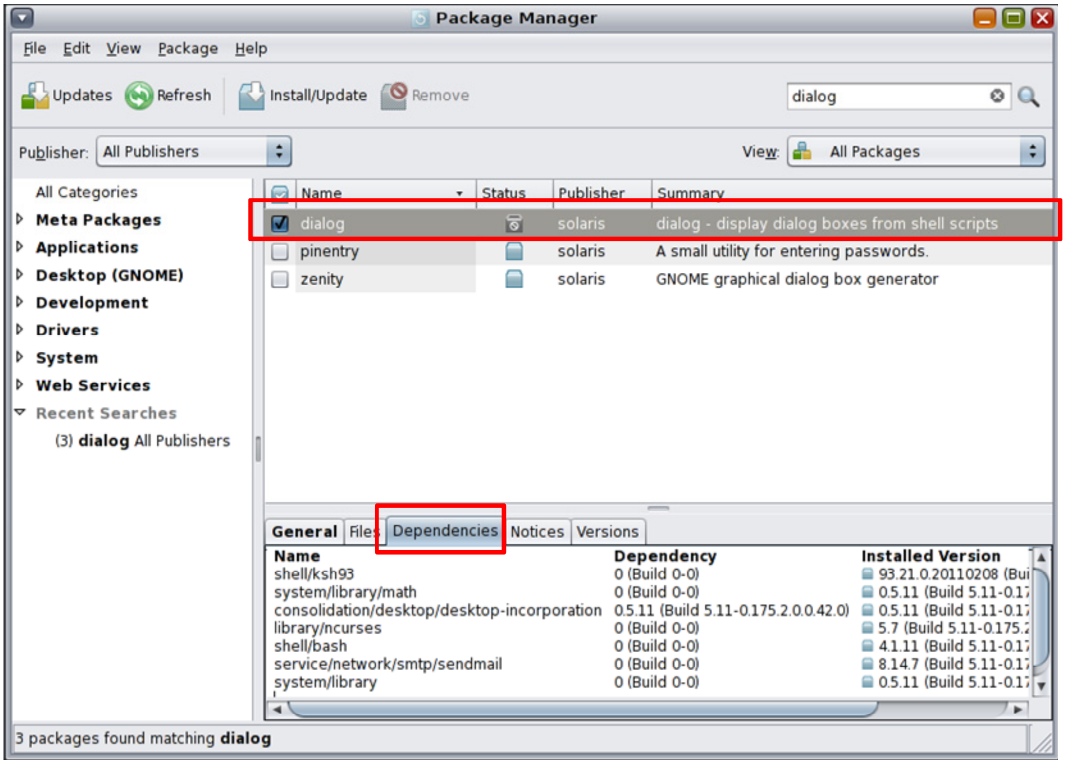
As you can see, the information differs slightly from the information displayed with the `pkg info` command.

Displaying the Files of a Package



With Package Manager, 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 slide screenshot). Here, you can see the files called out in the `dialog` manifest.

Displaying Package Dependency Information

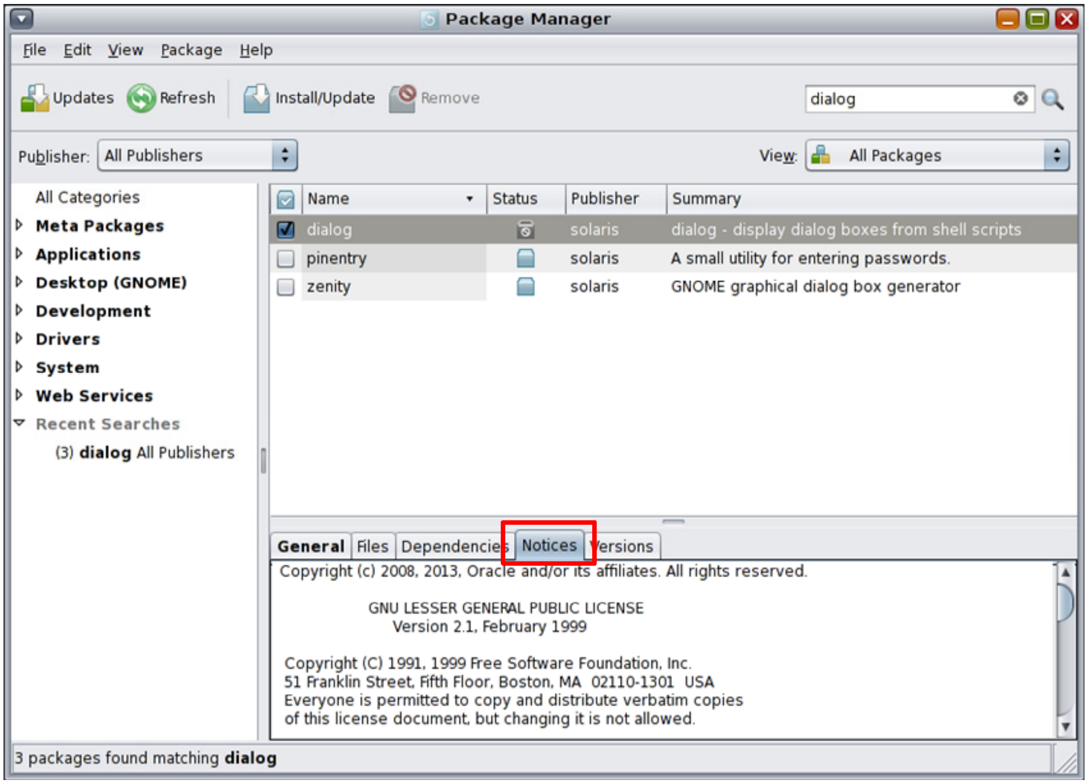


Software packages are often dependent on having other packages installed in order to run properly. By using Package Manager, you can display all package dependencies for a package by selecting it and clicking the Dependencies tab, as shown in the example in the slide.

The following information is displayed on the Dependency tab:

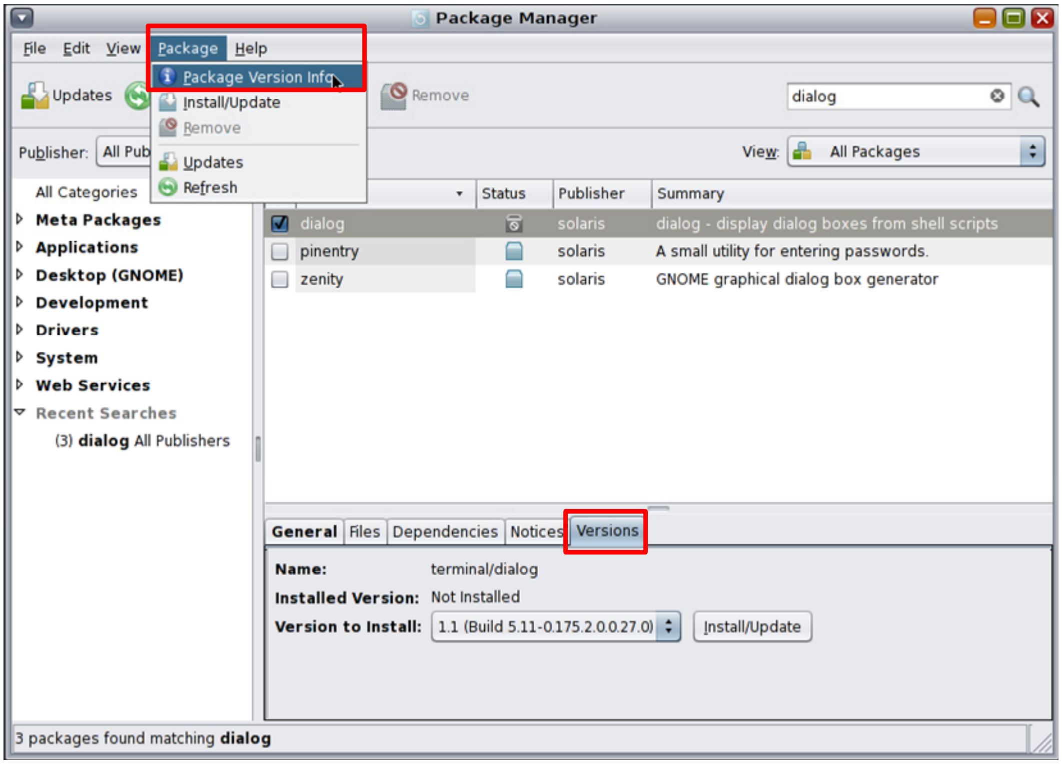
- **Name:** Name of the package
- **Dependency:** Build information
- **Installed Version:** The version of the package that is currently installed on the system

Displaying Package Notices



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

Displaying Package Versions



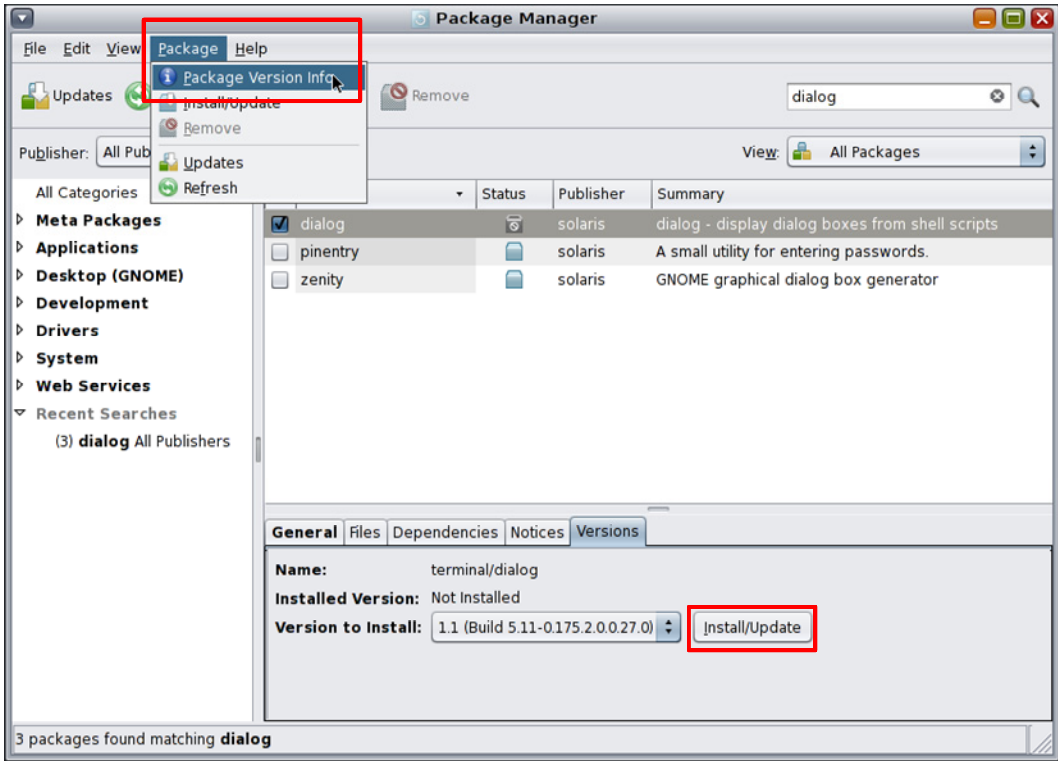
There are several ways in which you can display version information for a package with Package Manager. You can select **Package** > **Package Version Info** from the menu, as shown in the first highlighted section of the screenshot in the slide. Or you can select the package and click the Versions tab,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

The following information is displayed on the Versions tab:

- **Name:** Is the name of the package
- **Installed Versions:** Indicates the 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.

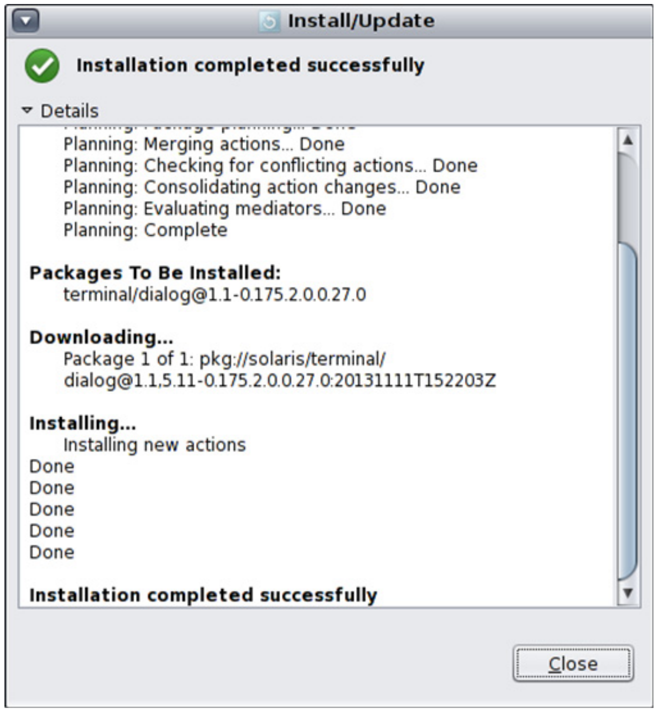
Note: The **Package** > **Package Version Info** option does not provide the **Install/Update** button.

Installing and Updating a Package



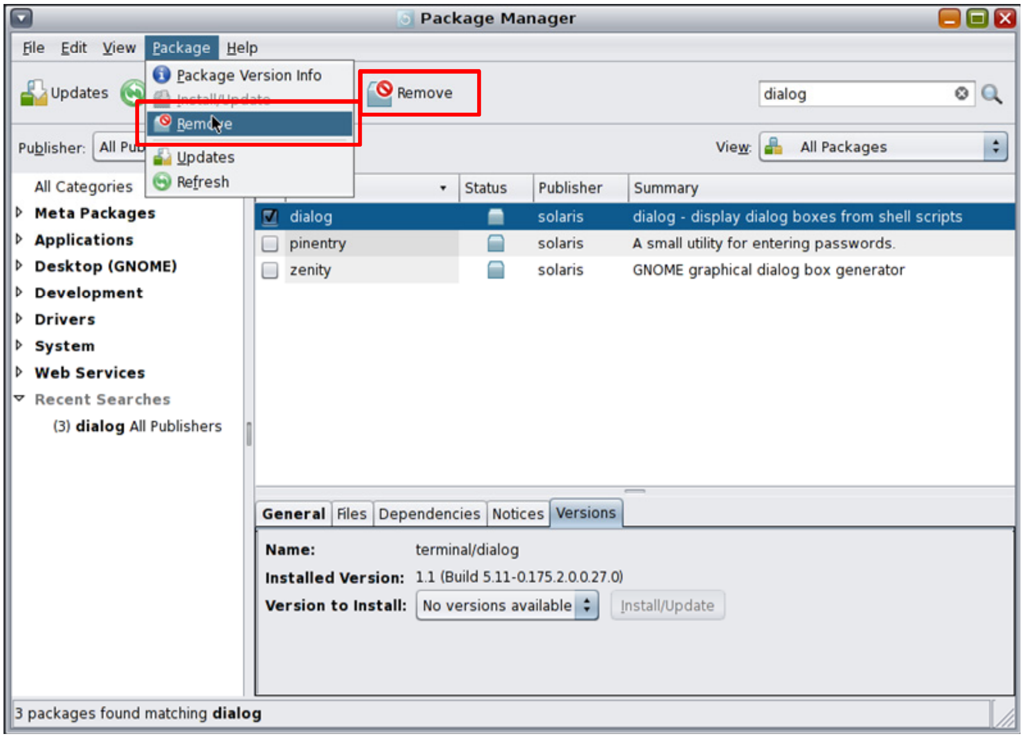
To install or update a package with Package Manager, you can either select **Package** > **Install/Update** from the menu or click the **Install/Update** button. As you can see in the example in the slide, a version of the package is available for installation.

Verifying a Package Installation



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

Uninstalling a Package



To uninstall a package by using Package Manager, highlight the package, and then either select **Package** > **Remove** from the menu or select the package and click the **Remove** button on the menu bar.

Practice 5-2 and Practice 5-3 Overview: Managing Software Packages by Using CLI 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 one hour to complete both these practices.

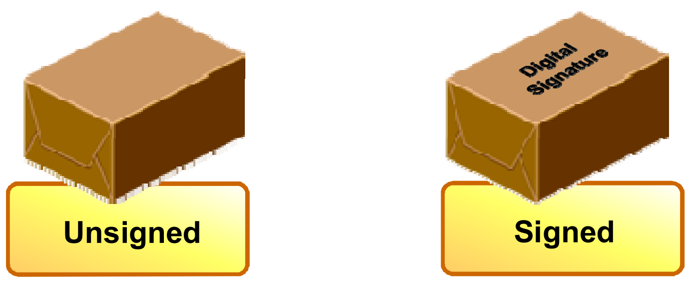
Lesson Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- Managing Software Packages
- **Managing Signed Packages and Package Properties**

Introducing Signed Packages

Signed packages contain digital signatures that verify that:

- The package came from the entity who signed it
- The entity signed the package
- The package has not been modified
- The entity is trusted



Software packages can be signed or unsigned. A signed package is identical to an unsigned package, except that it has a digital signature that verifies the following:

- The package came from the entity who signed it.
- The entity indeed signed it.
- The package has not been modified since the entity signed it.
- The entity who signed it is a trusted entity.

In other words, a signed package provides the reassurance that the package is secure and, therefore, safe to download and install on your system.

If you are interested in creating your package, and then publishing it to the repository, refer to http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-097-create-pkg-ips-524496.html.

Installing Signed Packages

- Identifying image properties for signed packages
- Configuring image properties for signed packages
- Identifying publisher properties for signed packages
- Configuring publisher properties for signed packages

Before you can install signed packages on your system, you must set specific properties on both the image and the publisher.

Identifying Image Properties for Signed Packages

Property	Description
`signature-policy`	Determines what checks will be performed on manifests when you install a package into the operating system image: • `ignore` • `verify` • `require-signatures` • `require-names`
`signature-required-names`	Defines names that must be seen as common names of certificates while validating the signatures of a package
`trust-anchor-directory`	Identifies the path name of the directory that contains the trust anchors for the image

ORACLE

You can set three image properties for signed packages. These are `signature-policy`, `signature-required-names`, and `trust-anchor-directory`.

- The `signature-policy` property, along with one of its values, determines the checks that are performed on manifests when installing a package into the operating system image. You can use the following values with this property:
  - **`ignore`:** Directs the image to ignore signatures for all manifests
  - **`verify`:** Directs the image to verify that all manifests with signatures are validly signed, but does not require all installed packages to be signed
  - **`require-signatures`:** Directs the image to require that all newly installed packages have at least one valid signature

    Note: The `pkg fix` and `pkg verify` commands also present a warning if an installed package does not have a valid signature.
  - **`require-names`:** Directs the image not only to require that all newly installed packages have at least one valid signature (as with the `require-signatures` property) but also to require that the strings listed in the `signature-required-names` image property appear as common names of the certificates that are used to verify the chains of trust of the signatures

- The next property, `signature-required-names`, defines a list of names that must be seen as common names of certificates while validating the signatures of a package. It is used only when the signature policy is `require-names`.
- The last property, `trust-anchor-directory`, identifies the path name of the directory that contains the trust anchors for the image. This path is relative to the operating system image.

The final policy that is applied to a package depends on the combination of image policy and publisher policy. The combination will be at least as strict as the stricter of the two policies taken individually.

Configuring Image Properties for Signed Packages

Use `pkg` with the following subcommands to configure package signature properties for an image:

- `set-property`
- `add-property-value`
- `remove-property-value`
- `unset-property`

Examples:

```
# pkg set-property signature-policy verify
# pkg add-property-value signature-require-names trustedname
# pkg remove-property-value signature-require-names trustedname
# pkg unset-property signature-policy
```


To configure image properties for signed packages, you use the `pkg` command with the list of subcommands presented in the slide. As you can see from the list of subcommands, you can set, add, remove, and unset properties.

Under “Examples” in the slide:

- The first command shows how to configure an image to verify all signed packages by setting the `verify` value for the `signature-policy` property.
- The second command is an example of how to add the string `trustedname` to the image’s list of common names that must be seen in a signature’s chain of trust for it to be considered valid.
- The third command is an example of how to remove the string `trustedname` from the image’s list of common names.
- The fourth command shows how to restore the `signature-policy` property to its default value.

Identifying Publisher Properties for Signed Packages

Property	Description
`signature-policy`	Determines the checks that will be performed on manifests when installing a package into the image from a specified publisher
`signature-required-names`	Defines names that must be seen as common names of certificates while validating the signatures of a package from a specified publisher

There are two publisher properties that you can set to use signed packages from a specific publisher: `signature-policy` and `signature-required-names`.

The `signature-policy` property is identical to the function of the `signature-policy` image property, except that this property applies only to packages from a specified publisher.

The `signature-required-names` property is identical to the function of the `signature-required-names` image property, except that this property applies only to packages from a specified publisher.

Configuring Publisher Properties for Signed Packages

Use `pkg set-publisher` with the following subcommands to configure package signature properties for a publisher:

- `--set-property`
- `--add-property-value`
- `--remove-property-value`
- `--unset-property`

Examples:

```
# pkg set-publisher --set-property signature-policy=require-
signatures whoisit.com
# pkg set-publisher --add-property-value signature-require-
names=trustedname whoisit.com
# pkg set-publisher --remove-property-value signature-require-
names=trustedname whoisit.com
# pkg set-publisher --unset-property signature-policy
whoisit.com
```


To configure publisher properties for signed packages, you use the `pkg set publisher` command with the list of subcommands presented in the slide. As with image properties, you can set, add, remove, and unset properties. Note that the subcommands are the same as those used for configuring image properties, with one exception: each subcommand is preceded by a double dash (`--`).

Under "Examples" in the slide, the first command shows how to configure an operating system image to require that all packages installed from the publisher `whoisit.com` must be signed.

The second command is an example of how to add the string `trustedname` to the `whoisit.com` publisher's list of common names that must be seen in a signature's chain of trust for it to be considered valid.

The third command is an example of how to remove the string `trustedname` from the `whoisit.com` publisher's list of common names.

The fourth command shows how to unset the `signature-policy` property that requires that all packages installed from the publisher `whoisit.com` must be signed.

Introducing Variants and Facets

- **Variant:** Mutually exclusive component of a package
  - Appears as a tag on IPS actions
  - Affects whether an IPS action is installable
- **Facet:** Optional component of a package
  - Appears as a tag on IPS actions
  - Affects whether an IPS action is installable

Another way that IPS allows you to manage your packages, regardless of whether they are signed or unsigned, is through the use of variants and facets. A variant is a mutually exclusive component of a package, such as architecture. Variants appear as tags on IPS actions and affect whether that action is installable. If an action has any variant tags, all variant tags must match the selection criteria to install the action.

Note: An action is defined in IPS as an installable object, such as a file, directory, link, or dependency. Actions are described in the manifest of a package. Every action consists primarily of its name and a key attribute. Together, these refer to a unique object as it follows a version history.

A facet is an optional component of a package, such as a locale. Facets appear as tags on IPS actions and affect whether that action is installable. If an action has any facet tags, at least one facet tag must match the selection criteria to install the action.

Displaying and Changing Variants and Facets

Variants

- To display the values of all variants, use `pkg variant`.
- To display specific variants, use `pkg variant` *`variant_spec`*.
- To change a variant, use `pkg change-variant --accept` *`variant_spec=instance`*.

Facets

- To display the current values of all facets defined in the current image, use `pkg facet`.
- To display specific facets, use `pkg facet` *`facet_spec`*.
- To change the current value of a facet, use *`pkg change-facet`* `--accept` *`facet_spec`*`=True|False|None`.

You can display and change the current values for both variants and facets. To display the values of all the variants of an image, you can use the `pkg variant` command. To display specific variants, use the `pkg variant` command with a variant specification (`variant_spec`). You can specify multiple variant specifications.

To change a specific variant, use the `pkg change-variant` command, followed by `--accept` and the variant specification (`variant_spec=instance`). If you want to see what will change without actually implementing the change, you can use the `-n` option as in the following example: `# pkg change-variant -n --accept variant.debug=false`.

To display all the facets in the current image, use `pkg facet`. To display specific facets, use the `pkg facet` command with a facet specification (`facet_spec`). You can specify multiple facet specifications. To change a specific facet, use the `pkg change-facet` command, followed by `--accept` (to automatically accept all package licenses without interruption to the update process) and the facet specification (`facet_spec=True|False|None`). Again, you can use the `-n` option to see what changes will occur without actually making the change.

Managing Package History

- To view package history, run `pkg history`.
- To view verbose package history information, run `pkg history -l`.
- To specify the number of the most recent package history operations to display, use the `-n` option.
- To display log records for a comma-separated list of time stamps, use the `-t` option.
- To purge package history, run `pkg purge-history`.

```
$ pkg history
START                  OPERATION          CLIENT              OUTCOME
2014-06-24T15:20:16    set-property     transfer module      Succeeded
2014-06-24T15:20:16    images-create    transfer module      Succeeded
2014-06-24T15:20:16    add-publisher    transfer module      Succeeded
…
# pkg purge-history
```


IPS has a `pkg history` command that enables you to see all package operations that have been performed on an image, as shown in the example in the slide.

Use the `-l` option if you want to display more information, including the outcome of the command, the time the command completed, the version and name of the client used, the name of the user who performed the operation, and any errors encountered while executing the command.

If you want to display only the most recent operations, use the `-n` option to specify the number of operations that you want to view.

To display log records for a comma-separated list of time stamps, use the `-t` option and a hyphen (`-`) between a start and finish time stamp. The keyword `now` can be used as an alias for current time.

Use the `pkg purge-history` command to purge history. However, you need root privileges to perform this action.

For more information, see man `pkg(1)`.

Quiz

Which command enables you to configure your current image to ensure that all manifests with signatures are validly signed?

a. `# pkg set-property signature-policy verify`

b. `# pkg set-property signature-policy require-names`

c. `# pkg set-property signature-policy require-signature`

ORACLE

Answer: a

Quiz

Which `pkg` subcommand or option of the `set-publisher` subcommand is used to configure publisher properties for signed packages?

a. `set-property`
b. `set-publisher`
c. `set-publisher property`
d. `--set-publisher`
e. `--set-property`

Answer: e

Summary

In this lesson, you should have learned how to:

- Describe IPS, its components, and interfaces
- Configure an IPS client to access the local IPS repository
- Manage package publishers
- Manage software packages
- Manage signed packages and package properties

6

Managing Data by Using ZFS

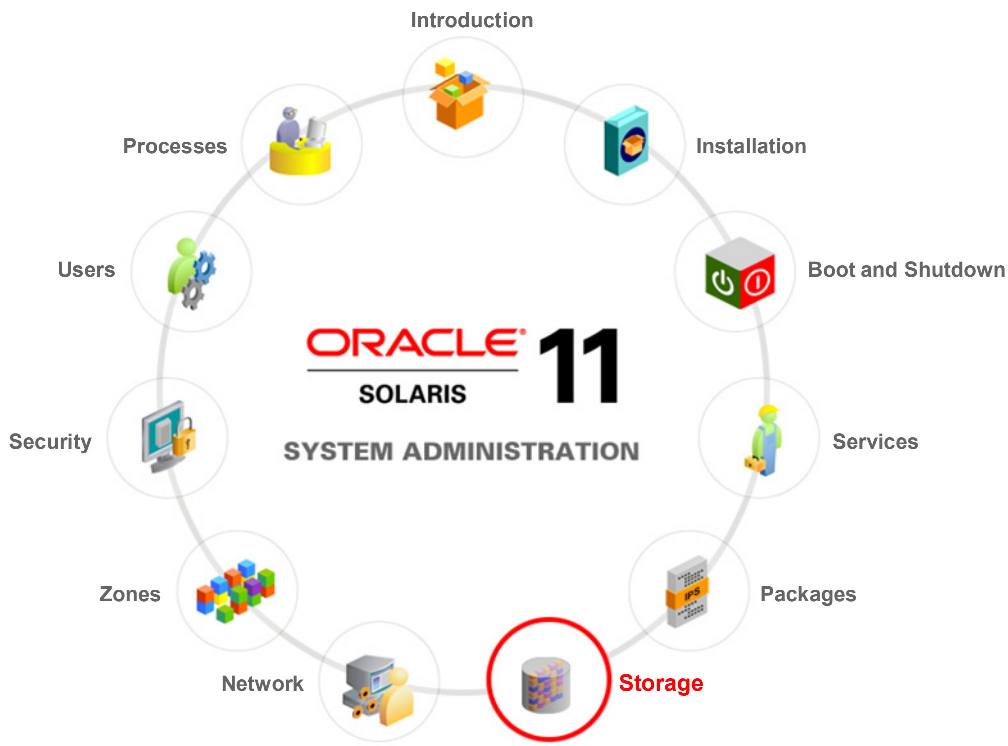
ORACLE

Objectives

After completing this lesson, you should be able to:

- Explain the role of ZFS in data management
- Administer ZFS storage pools
- Administer ZFS file systems
- Administer ZFS properties
- Administer ZFS 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, tested the system's boot and shutdown functionality, as well as SMF services. Now you will 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. 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 administering ZFS storage pools, file systems, properties, snapshots, and clones.

Lesson Agenda

- **Introducing ZFS**
- Administering ZFS Storage Pools
- Administering ZFS File Systems
- Administering ZFS Properties
- Administering ZFS Snapshots and Clones

Importance of Data Management

Data management 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 proper storage and management of business application data is extremely important for your company's continued success, it is necessary to understand Oracle Solaris 11's dynamic storage capabilities. You will need 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.

Furthermore, backing up and restoring data is covered in the *Oracle Solaris 11 Advanced System Administration* course, which is a follow-on to this course.

Introduction to ZFS

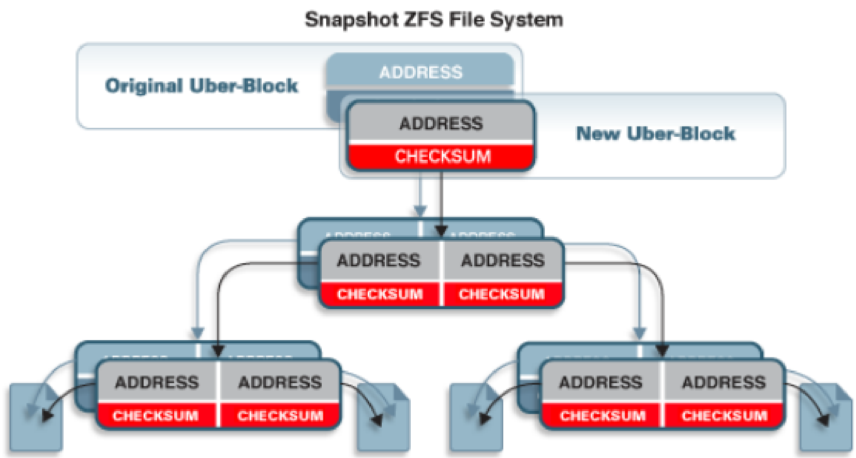
- Two products in one:
  - Volume manager
  - File system
- Most scalable file system ever:
  - 128-bit file system
  - Up to 256 trillion directory entries allowed
  - No limit to the number of file systems or number of files contained within a file system
- Transactional file system:
  - File system 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.

Several file system types are available for configuring storage in Oracle Solaris, but ZFS is the default file system used in Oracle Solaris 11. Before Oracle Solaris 10, the traditional type of file system that was used for storing data in the Oracle Solaris environment was the UFS. The ZFS file system was introduced in Oracle Solaris 10.

ZFS takes a completely new approach to disk storage space management. It revolutionizes the traditional file systems that were used in previous versions of Oracle Solaris, or UNIX for that matter. ZFS does not replace those traditional file systems, nor is it an improvement on that existing technology. Rather, it is a fundamentally new approach to data management. ZFS was designed to be more robust, more scalable, and easier to administer than traditional Oracle Solaris file systems. One of the goals 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.

ZFS employs a copy-on-write (CoW) methodology. Data is never overwritten in place. The following diagram shows the creation of new data blocks with modified data in them rather than the overwriting of existing data blocks that the new data modifies. ZFS then creates new pointers to the new data and a new master block (uberblock) that points to the modified data tree. Only then does it move to using the new uberblock and tree. In addition to providing data integrity, having new and previous versions of the data on disk allows for services such as snapshots to be implemented very efficiently.



Moreover, ZFS is self-healing transactional file system, which means that 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 ensures 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 is always consistent. In addition, synchronous data is always guaranteed to be written before returning, so it is never lost.

In a ZFS file system, a checksum is computed for every block to prevent silent data corruption. A checksum is a value that is used to ensure that data is stored without error. It is derived by calculating the binary value in a block of data by using a particular algorithm and storing the calculated results with the data. When data is retrieved, the checksum is recalculated and matched against the stored checksum. If the checksums are the same, the data has not changed. If the checksums are different, the data has been changed, corrupted, or tampered with.

ZFS Terms

Term	Description
Checksum	A 256-bit hash of the data in a file system block
Clone	A file system with contents that are identical to the contents of a ZFS snapshot
Dataset	A generic name for ZFS entities such as clones, file systems, snapshots, and volumes
Resilvering	The process of transferring data from one device to another
Scrub	A tool that validates and repairs the ZFS file system (including the metadata) while the file system is online and mounted
Snapshot	A read-only image of a file system or volume at a given point of time
zpool	A ZFS dataset that is mounted within the standard system namespace and that behaves like other traditional file systems

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

Historically, file systems were constructed on top of a single physical device. Disks had to be added to the system, and then divided into one or more file systems. As you add data to a file system, the file system begins to fill up. In case you require more space, you had to manually allocate more space to that file system. In such scenarios, chances are that you allocate too much space to one file system while another file system fills up, or you add another disk, or take away space from another file system to get more free space. Taking away space from an existing file system typically requires backing up, destroying, and rebuilding the existing file system.

ZFS uses the concept of storage pools, called *zpools*, 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. File systems are no longer constrained to individual devices, thereby allowing them to share disk space with all the file systems in the pool. You no longer need to predetermine the size of a file system, because file systems grow automatically within the disk space that is allocated to the storage pool. When new storage is added, all file systems within the pool can immediately use the additional disk space without additional work.

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 that is at least 128 MB in size
- Typically, a hard drive that is 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 that is 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. 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.

ZFS formats the disk by using an Extensible Firmware Interface (EFI) label to contain a single, large slice. In this course, you use whole disks only. On an Oracle Solaris 11.2 system, the partition table that is displayed by the `format` command appears similar to the following:

```
Current partition table (original):
Total disk sectors available: 33537981 + 16384 (reserved sectors)
Part      Tag   Flag First Sector     Size     Last Sector
0  BIOS_boot   wm            256   256.00MB      524543
1        usr   wm         524544    15.74GB      33538014
2 unassigned   wm              0          0            0
3 unassigned   wm              0          0            0
4 unassigned   wm              0          0            0
5 unassigned   wm              0          0            0
6 unassigned   wm              0          0            0
8   Reserved   wm       33538015 8.00MB            33554398
```

In the preceding output, partition 0 (`BIOS boot`) contains the required GUID Partition Table (GPT) boot information. Similar to partition 8, it requires no administration and should not be modified. The root file system is contained in partition 1.

A SPARC system with GPT-aware firmware has an EFI (GPT) disk label applied. Example:

```
Current partition table (original):
Total disk sectors available: 143358320 + 16384 (reserved sectors)
Part      Tag   Flag First Sector      Size    Last Sector
0        usr   wm            256    68.36GB    143358320
1 unassigned   wm              0        0           0
2 unassigned   wm              0        0           0
3 unassigned   wm              0        0           0
4 unassigned   wm              0        0           0
5 unassigned   wm              0        0           0
6 unassigned   wm              0        0           0
8   reserved   wm      143358321     8.00MB    143374704
```

ZFS Storage Pool Components: Disks

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

ORACLE

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 an 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.

Review the following considerations when using whole disks in your ZFS storage pools:

- When using a whole disk, the disk is generally named by using the `/dev/dsk/cXtXdX` naming convention. Some third-party drivers use a different naming convention or place disks in a location other than the `/dev/dsk` directory. To use these disks, you must manually label the disk and provide a slice to ZFS.
- On an x86-based system, the disk must have a valid Solaris `fdisk` partition.
- ZFS applies an EFI label when you create a storage pool with whole disks.
- The Oracle Solaris installer applies an EFI (GPT) label for the root pool disks on a SPARC-based system with GPT-aware firmware and on an x86-based system, in most cases.

ZFS Storage Pool Components: Slices

- Disks can be labeled with an SMI label.
- For the bootable ZFS root pool:
  - A disk must contain slices
  - An SMI label is required

Examples

- On a SPARC-based system with 72-GB disk:
  - 68 GB of usable space in slice 0
- On an 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
    - Not changeable

ORACLE

Disks can be labeled with a traditional Solaris VTOC (SMI) label when you create a storage pool with a disk slice but using disk slices for a pool is not recommended because management of disk slices is more difficult.

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.

```
# format
.
.
.
Specify disk (enter its number): <disk number>
selecting c1t1d0
partition> p
Current partition table (original):
```

```
Total disk cylinders available: 14087 + 2 (reserved cylinders)
Part      Tag Flag Cylinders    Size          Blocks
0        root wm   0 - 14086 68.35GB (14087/0/0) 143349312
1 unassigned wm   0              0   (0/0/0)            0
2      backup wm   0 - 14086 68.35GB (14087/0/0) 143349312
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
```

Similarly, on an 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.

```
# format
.
.
.
selecting c1t0d0
partition> p
Current partition table (original):
Total disk cylinders available: 49779 + 2 (reserved cylinders)
Part      Tag Flag Cylinders    Size          Blocks
0        root wm   1 - 49778 68.36GB (49778/0/0) 143360640
1 unassigned wu   0             0    (0/0/0)            0
2      backup wm   0 - 49778 68.36GB (49779/0/0) 143363520
3 unassigned wu   0             0    (0/0/0)            0
4 unassigned wu   0             0    (0/0/0)            0
5 unassigned wu   0             0    (0/0/0)            0
6 unassigned wu   0             0    (0/0/0)            0
7 unassigned wu   0             0    (0/0/0)            0
8        boot wu   0 - 0      1.41MB (1/0/0)         2880
9 unassigned wu   0             0    (0/0/0)            0
```

Note: An `fdisk` partition also exists on an x86-based system. An `fdisk` partition is represented by a `/dev/dsk/cN[tN]dNpN` device name, and acts as a container for the disk's available slices. Do not use a `cN[tN]dNpN` device for a ZFS storage pool component because this configuration is neither tested nor supported.

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

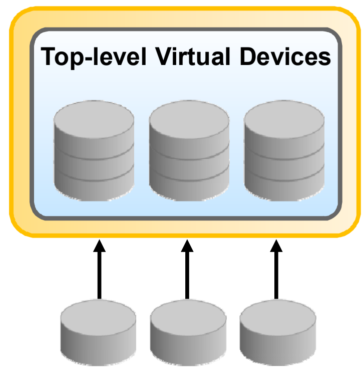
ZFS also allows you to use UFS files as virtual devices in your storage pool. This feature is not intended for production use, however. It is aimed primarily at testing and enabling simple experimentation. The reason is that any use of files relies on the underlying file system for consistency. If you create a ZFS pool that is 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 the top of a 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 consists 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 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 the time of “write.”
- 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.

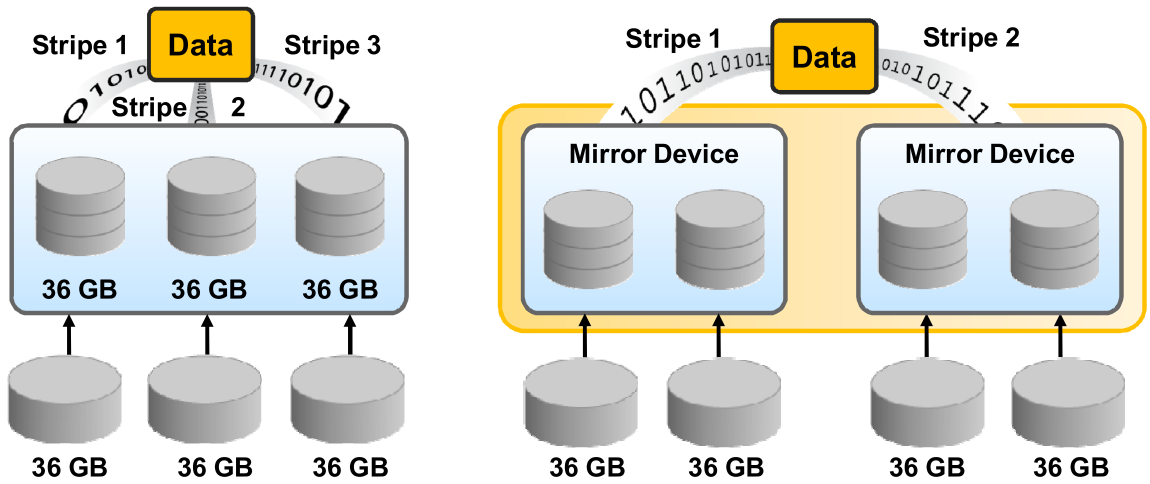
ZFS dynamically stripes data across all top-level virtual devices. The decision about where to place data is done at the time of “write,” so no fixed-width stripes are created at allocation time.

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 the top-level virtual devices.



Stand-alone Devices | **Mirrored Devices**

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

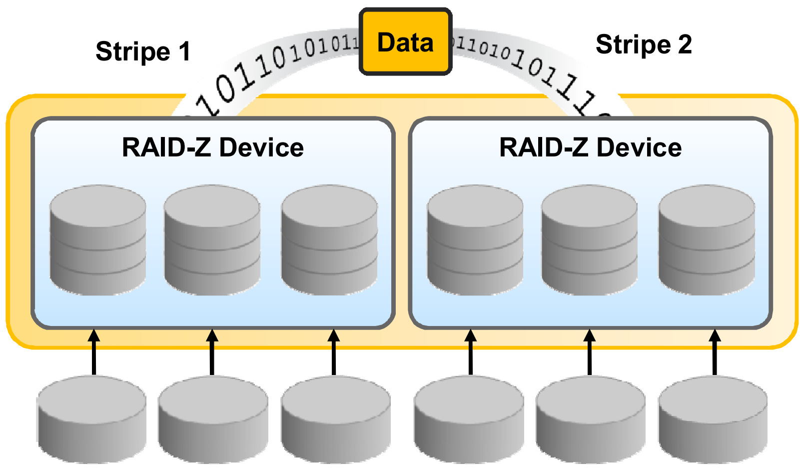
The diagram on the left shows how data is striped across three stand-alone disks, with each top-level virtual device mapped to a 36 GB disk. A ZFS pool that uses disks as top-level virtual devices, as shown in this diagram, provides no data redundancy.

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, 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 in the slide, 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.

You learn more about this topic in the follow-on *Oracle Solaris 11 Advanced System Administration* course.

ZFS Storage Pool Types

Storage Pool Type	Description
Basic Storage Pool	A storage pool with a configuration that has a minimum of one disk
Mirrored Storage Pool	A storage pool with a configuration that has at least two disks, preferably on separate controllers. Many disks can be used in a mirrored configuration.
RAID-Z Storage Pool	A storage pool that consists of 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.

ZFS File Systems

- Are default disk-based and root file systems in Oracle Solaris 11
- Share space with all the file systems in the pool
- Grow automatically within the space allocated to the storage pool
- Immediately use additional space when new storage is added

A *file system* is a structure of directories that is used to organize and store files. The term file system is used to describe the following:

- A particular type of file system (disk-based, network-based, or virtual)
- The entire file tree, beginning with the root (`/`) directory
- The data structure of a disk slice or other media storage device
- A portion of a file tree structure that is attached to a mount point on the main file tree so that the files are accessible

The ZFS file system is hierarchical, starting with the root directory (`/`) and continuing downward through a number of directories. The Oracle Solaris installation process enables you to install a default set of directories, and uses a set of conventions to group similar types of files together. For a brief overview of Oracle Solaris file systems and directories, see the `filesystem`(5) man page.

File systems are no longer constrained to individual devices; that is, they can share space with all the file systems in the pool. Further, 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 the file systems in the pool can immediately use the additional space.

Directory Structure of ZFS File System

File System or Directory	Description
root (/)	Is the top of the hierarchical file tree and contains directories and files that are critical for system operation
/dev	Contains device files, which include terminal devices, USB, or any device attached to the system
/etc	Contains configuration files required by all programs
/usr	Contains system files and directories that can be shared with other users
/export/home or /home	Is the mount point for user home directories, which store user work files. By default, the /home directory is an automounted file system.
/var	Includes system files and directories that are likely to change or grow over the life of the local system. These include system logs.
/opt	Is the optional mount point for third-party software

The table in the slide provides a brief list of directories of the ZFS file system.

Managing Data

As part of learning how to manage data, you will learn about implementing the following in the next section:

- 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 management needs of your company, it is time for you explore 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 section, you learn the commands that you need to perform these tasks.

Lesson Agenda

- Introducing ZFS
- **Administering ZFS Storage Pools**
- Administering ZFS File Systems
- Administering ZFS Properties
- Administering ZFS Snapshots and Clones

Determining Your ZFS Storage Pool Requirements

As part of data management, you should identify your storage pool device requirements:

- Disks that are at least 128 MB in size
- Disks not in use by other parts of the operating system
- Entire disks that are formatted as a single, large slice or individual slices on a preformatted disk

To set up your ZFS environment, you must first create a ZFS storage pool based on the storage requirements that were 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 data must be identified. The devices that are selected must meet the following criteria. They must be disks of at least 128 MB size, and 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.

Creating ZFS Storage Pools

- Use the `zpool create` command to create a basic storage pool.
- The `zpool create` command:
  - Accepts a pool name
  - Accepts any number of virtual devices

In this section, you 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 as 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

In the example command in the slide, a pool called `hrpool` is created by using the `zpool create` command. 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.

The `zpool` command is enhanced to support an EFI (GPT) disk label so that if you need to re-create a root pool after the system is installed, you can do so with the `zpool create -B` command. This new command option creates the required partitions and boot information that is needed for booting.

Determining Local Storage Disk Availability

To display disk availability, run `format`.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <ATA-VBOX HARDDISK-1.0-40.00GB>
          /pci@0,0/pci8086,2829@d/disk@0,0
       1. c1t2d0 <ATA-VBOX HARDDISK-1.0 cyl 1022 alt 2 hd 64 sec 32>
          /pci@0,0/pci8086,2829@d/disk@2,0
       2. c1t3d0 <ATA-VBOX HARDDISK-1.0 cyl 1022 alt 2 hd 64 sec 32>
          /pci@0,0/pci8086,2829@d/disk@3,0
       3. c1t4d0 <ATA-VBOX HARDDISK-1.0 cyl 1022 alt 2 hd 64 sec 32>
          /pci@0,0/pci8086,2829@d/disk@4,0
       4. c1t5d0 <ATA-VBOX HARDDISK-1.0 cyl 1022 alt 2 hd 64 sec 32>
          /pci@0,0/pci8086,2829@d/disk@5,0
       5. c1t6d0 <ATA-VBOX HARDDISK-1.0 cyl 1022 alt 2 hd 64 sec 32>
           /pci@0,0/pci8086,2829@d/disk@7,0
       8. c1t7d0 <ATA-VBOX HARDDISK-1.0 cyl 1022 alt 2 hd 64 sec 32>
           /pci@0,0/pci8086,2829@d/disk@8,0
       7. c1t8d0 <ATA-VBOX HARDDISK-1.0 cyl 1022 alt 2 hd 64 sec 32>
           /pci@0,0/pci8086,2829@d/disk@9,0
```


You can determine the disks that 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, `c1t0d0`, 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. The senior system or storage administrators in your organization are responsible for formatting the disks (if necessary) and making them available on the system.

If, by chance, you select 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, refer to the following URLs:

- http://docs.oracle.com/cd/E36784_01/html/E36834/disksconcepts-29477.html#scrolltoc
- http://docs.oracle.com/cd/E36784_01/html/E36834/disksprep-31030.html#scrolltoc

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 c1t3d0 c1t4d0 mirror c1t5d0 c1t6d0
```

Data is:

- Dynamically striped across both mirrors
- Redundant between each disk within a mirror

A mirrored storage pool is created by 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 in the slide creates a pool called `hrpool` with two 2-way mirrors. The first mirror contains the devices `c1t3d0` and `c1t4d0`, and the second mirror contains the devices `c1t5d0` and `c1t6d0`.

Data is dynamically striped across both mirrors, with data being redundant between each disk within a mirror.

Creating a ZFS Root Pool

To create a ZFS root pool, enter `zpool create`, followed by the `-B` option to create a boot partition, the root pool name, and the disk name to be included in the pool.

```
# zpool create -B rpool2 c1t2d0
```


Starting with Oracle Solaris 11.1 release, disks that are used for the root pool are installed with an EFI (GPT) label on an x86-based system, a supported SPARC system with GPT-aware firmware. Or, an SMI (VTOC) label is applied on a SPARC-based system without GPT-aware firmware. The installer applies an EFI (GPT) label if possible and if you need to re-create a ZFS root pool after installation, you can use the following command to apply the EFI (GPT) disk label and the correct boot information:

```
# zpool create -B rpool2 c1t2d0
```

The root pool must be created as a mirrored configuration or as a single-disk configuration. The `-B` option, when operating on a whole disk device, creates the boot partition, if one is required to boot from the EFI (GPT) labeled disks on the platform. The `-B` option has no effect on devices that are not whole disks.

Note: A RAID-Z or a striped configuration is not supported.

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 c1t2d0 c1t3d0 c1t4d0 c1t5d0
/dev/dsk/c1t6d0
```

```
# zpool create datapool raidz2 c1t2d0 c1t3d0 c1t4d0 c1t5d0
c1t6d0 c1t7d0 raidz2 c1t8d0 c1t9d0 c1t10d0 c1t11d0 c1t12d0
c1t13d0
```


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`, is created by using the `zpool create` command. The pool consists of five disks: `c1t2d0`, `c1t3d0`, `c1t4d0`, `c1t5d0`, and `c1t6d0`. The `/dev/dsk/` path is included for the `c1t6d0` disk to illustrate that disks can be specified by using their full paths. The `/dev/dsk/c1t6d0` device is identical to the `c1t6d0` device.

You could also use disk slices in this configuration, but you would need to pre-format the disks to have appropriately sized slices.

In the second code example, a double-parity RAID-Z storage pool, called `datapool`, is created. The first `raidz2` virtual device contains six disks: `c1t2d0`, `c1t3d0`, `c1t4d0`, `c1t5d0`, `c1t6d0`, and `c1t7d0`. The second `raidz2` virtual device also contains six disks: `c1t8d0`, `c1t9d0` , `c1t10d0` , `c1t11d0` , `c1t12d0`, and `c1t13d0`.

Default Mount Point for Storage Pools

- The default mount point is */pool-name*.
- A directory is automatically created if it does not exist.
- If a directory exists, it must be empty.

```
# zpool create home c1t2d0
default mountpoint '/home' exists and is not empty
use '-m' option to provide a different default
```

- To change the default mount point, use `-m` with `zpool create`.

```
# zpool create -m /export/zfs home c1t2d0
```


When a pool is created, the default mount point for the root dataset is */pool-name*. This directory must either 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 get an error message stating that the mount point exists and that it is not empty. The error message directs 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 in the slide, 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 by using the `zpool create` command, spend 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 that 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 in the slide, the pool named `testpool` is being destroyed.

Note: For more information about recovering a pool, refer to http://docs.oracle.com/cd/E36784_01/html/E36835/gcfhw.html#scrolltoc.

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 property information for a pool.

```
# zpool get all hrpool
NAME    PROPERTY       VALUE                 SOURCE
hrpool  allocated      85K                   -
hrpool  altroot        -                     default
hrpool  autoexpand     off                   default
hrpool  autoreplace    off                   default
hrpool  bootfs         -                     default
hrpool  cachefile      -                     default
hrpool  capacity       0%                    -
hrpool  dedupditto     0                     default
hrpool  dedupratio     1.00x                 -
hrpool  delegation     on                    default
hrpool  failmode       wait                  default
hrpool  free           15.9G                 -
hrpool  guid           13211416720083688767  -
hrpool  health         ONLINE                -
hrpool  listshares     off                   default
hrpool  listsnapshots  off                   default
hrpool  readonly       off                   -
hrpool  size           15.9G                 -
hrpool  version        35                    default
```

ORACLE

You can display all property information for a pool by using the `zpool get all` command, followed by the pool name.

In the example shown in the slide, property information is being displayed for `hrpool`. 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 see most often in this class. For the most complete and up-to-date listing and a description of the ZFS pool properties, refer to http://docs.oracle.com/cd/E36784_01/html/E36835/gfifk.html#scrolltoc.

Displaying Pool Properties

```
# zpool get all hrpool
NAME    PROPERTY       VALUE                 SOURCE
hrpool  allocated      85K                   -
hrpool  altroot        -                     default
hrpool  autoexpand     off                   default
hrpool  autoreplace    off                   default
hrpool  bootfs         -                     default
hrpool  cachefile      -                     default
hrpool  capacity       0%                    -
hrpool  dedupditto     0                     default
hrpool  dedupratio     1.00x                 -
hrpool  delegation     on                    default
hrpool  failmode       wait                  default
hrpool  free           15.9G                 -
hrpool  guid           13211416720083688767  -
hrpool  health         ONLINE                -
hrpool  listshares     off                   default
hrpool  listsnapshots  off                   default
hrpool  readonly       off                   -
hrpool  size           15.9G                 -
hrpool  version        35                    default
```


Observe in the slide that four properties are highlighted: `autoreplace`, `capacity`, `health`, and `size`.

- **`autoreplace`:** You have already learned a little bit about this property. 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`.
- **`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`. In the example, `capacity` is `0%`.
- **`health`:** This read-only property identifies the current health of the pool as `ONLINE`, `DEGRADED`, `FAULTED`, `OFFLINE`, `REMOVED`, or `UNAVAIL`. In this example, `hrpool` has a health status of `ONLINE`.
- **`size`:** This is a read-only property that identifies the total size of the storage pool. In this example, `hrpool` pool is `15.9G` in size.

Displaying Pool Properties

```
# zpool get all hrpool
NAME    PROPERTY       VALUE                SOURCE
hrpool  allocated      85K                  -
hrpool  altroot        -                    default
hrpool  autoexpand     off                  default
hrpool  autoreplace    off                  default
hrpool  bootfs         -                    default
hrpool  cachefile      -                    default
hrpool  capacity       0%                   -
hrpool  dedupditto     0                    default
hrpool  dedupratio     1.00x                -
hrpool  delegation     on                   default
hrpool  failmode       wait                 default
hrpool  free           15.9G                -
hrpool  guid           13211416720083688767 -
hrpool  health         ONLINE               -
hrpool  listshares     off                  default
hrpool  listsnapshots  off                  default
hrpool  readonly       off                  -
hrpool  size           15.9G                -
hrpool  version        35                   default
```

ORACLE

Observe in the slide that four properties are highlighted: `allocated`, `autoexpand`, `free`, and `listsnapshots`.

- **`allocated`:** It 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 85 KB of space.
- **`autoexpand`:** By using this 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.
- **`free`:** This is a read-only value that identifies the amount of storage that is available within the pool. In this example, there is 15.9 GB of space available in the pool that is named `hrpool`.
- **`listsnapshots`:** This property controls whether the snapshot information that is associated with this pool is displayed with the `zfs list` command. The default value is `off`. You look at snapshots in greater detail in the lesson titled "Managing Data by Using ZFS." In this example, the `listsnapshots` property is set to the default, `off`.

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 about pool status. The information that you can obtain generally falls into three categories: basic pool usage information, pool I/O statistics, and pool health status. You can also display command history for a pool.

In this section, you learn about 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 the pools in 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 the pools in the system.

The example in the slide shows two pools in 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 (`dedup`) property controls the ability to remove duplicate data in a ZFS file system. If a file system has the `dedup` property enabled, duplicate data blocks are removed synchronously. The result is that only unique data is stored, and common components are shared between files. You can enable this property as follows:

```
# zfs set dedup=on hrpool/home
```

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
```

ORACLE

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 the example in the slide).

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 in the slide, the `-H` option is 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 accumulated statistics since boot for all the pools in 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. 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 accumulated statistics since boot for all the pools in the system.

In this example, the `zpool iostat` command is 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, which is 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, refer to http://docs.oracle.com/cd/E36784_01/html/E36835/gammt.html#scrolltoc.

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
pool           alloc  free      read  write     read  write
----------     -----  -----     -----  -----    -----  -----
hrpool         20.4G  59.6G         0     22        0   6.00
mirror         20.4G  59.6G         0     22        0   6.00
  c1t3d0           -      -         1    295    11.2K    148
  c1t4d0           -      -         1    299    11.2K    148
----------     -----  -----     -----  -----    -----  -----
```


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 in the slide, look at the I/O statistics for the virtual devices in the mirrored pool named `hrpool`: `c1t3d0` and `c1t4d0`. 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.
  - This is most noticeable immediately after pool creation.
  - It should gradually equalize.
  - Broken, unresponsive, or offlined devices can affect symmetry as well.

Note: You can also specify intervals when examining virtual device statistics.

ORACLE

When viewing I/O statistics on a virtual device basis, there are two things that you need to remember.

- Space usage is available for top-level virtual devices only.
- 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, because a significant amount of I/O is performed directly on 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 that is 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

A pool's health status is described by one of the following four states:

State	Description
DEGRADED	A pool with one or more failed devices, whose data is still available due to a redundant configuration
ONLINE	A pool that has all devices operating normally
SUSPENDED	A pool that is waiting for device connectivity to be restored. A SUSPENDED pool remains in the wait state until the device issue is resolved.
UNAVAIL	A pool with corrupted metadata, or one or more unavailable devices, and insufficient replicas to continue functioning

Determining the Health Status of a Pool

Each device can fall into one of the following states:

State	Description
`ONLINE`	The device or virtual device is in normal working order.
`DEGRADED`	The virtual device has experienced failure but is still able to function.
`OFFLINE`	The device has been explicitly taken offline by the administrator.
`REMOVED`	The device was physically removed while the system was running.
`UNAVAIL`	The device or virtual device cannot be opened (that is, it is unavailable).

Note: The `DEGRADED` state is most common when a mirror or RAID-Z device has lost one or more constituent devices. The fault tolerance of the pool might be compromised, because a subsequent fault in another device might be unrecoverable.

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 `UNAVAIL`	`DEGRADED`	The pool continues to run but the level of redundancy or data throughput might be affected.
One or more vdevs `UNAVAIL` or `OFFLINE`	`UNAVAIL` or `SUSPENDED`	The pool is completely inaccessible. No data recovery is possible until devices are attached or repaired.

You can recall that a virtual device (vdev) is a logical device in a pool, which can be a physical device, a file, or a collection of devices. The health of a pool is determined from the health of all its top-level virtual devices.

If all virtual devices are `ONLINE`, the pool is also `ONLINE`.

If any one of the virtual devices is `degraded` or `unavailable`, 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 `unavailable` or `offline`, the pool is also in the `unavailable` or `suspended` state. A pool in the `unavailable` or `suspended` 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 in 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
  scan: none requested
config:
        NAME            STATE     READ  WRITE  CKSUM
        hrpool          ONLINE       0      0      0
          mirror-0      ONLINE       0      0      0
            c1t3d0      ONLINE       0      0      0
            c1t4d0      ONLINE       0      0      0

errors: No known data errors
```


You can examine the health of a specific pool by specifying a pool name after 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 the example in the slide.

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: hrpool1
 state: DEGRADED
status: One or more devices are unavailable in response to persistent errors.
Sufficient replicas exist for the pool to continue functioning in a degraded state.
action: Determine if the device needs to be replaced, and clear the errors using
'zpool clear' or 'fmadm repaired', or replace the device with 'zpool replace'.
Run 'zpool status -v' to see device specific details.
  scan: none requested
config:
NAME         STATE     READ WRITE CKSUM
hrpool1      DEGRADED     0     0     0
  mirror-0   DEGRADED     0     0     0
    c1t3d0   ONLINE       0     0     0
    c1t4d0   UNAVAIL      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 in the slide.

Notice that in this example, the mirrored pool named `hrpool` is in degraded state because the device `c1t0d0` has faulted and cannot be opened.

Notice also that according to the status message, although the pool is in 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 by using the `zpool online` command, and the URL after `see`, which provides additional information about how to resolve the issue.

By 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 hrpool
History for 'hrpool':
2013-10-24.11:20:57 zpool create hrpool c1t3d0 c1t4d0
```


Now, take a brief look at how to display 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 the example in the slide).

From the output in the example, you can see that on October 24, 2013, at 11:20:57, the pool named `hrpool` was created by using two disks: `c1t3d0` and `c1t4d0`.

In case you forget any of the `zpool` subcommands and their syntaxes, the `zpool` command has a `help` subcommand that you can use to view more information about all its subcommands and their supported options. Example:

```
# zpool help
The following commands are supported:
add      attach   clear   create  destroy  detach   export  get
help     history  import  iostat  list     offline  online  remove
replace  scrub    set     split   status   upgrade
For more info, run: zpool help <command>
# zpool help attach
usage:
attach [-f] <pool> <device> <new-device>
```

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

Which 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

Which 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 `DEGRADED` state, the data is completely inaccessible.

a. True

b. False

Answer: b

Practice 6-1 and Practice 6-2 Overview: Administering ZFS Storage Pools

In practice 6-1, you perform the following tasks on a whole disk:

- Creating different types of ZFS pools
- Querying the pool attributes

In practice 6-2, you perform the following tasks on disk slices:

- Formatting a disk
- Creating a ZFS pool
- Examining the pool's configuration
- Creating a ZFS file system
- Deleting the pool

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

- **Practice 6-1:** Administering ZFS storage pools
- **Practice 6-2:** Administering ZFS pools by using disk slices
- **Practice 6-3:** Administering ZFS file systems
- **Practice 6-4:** Administering ZFS snapshots and clones

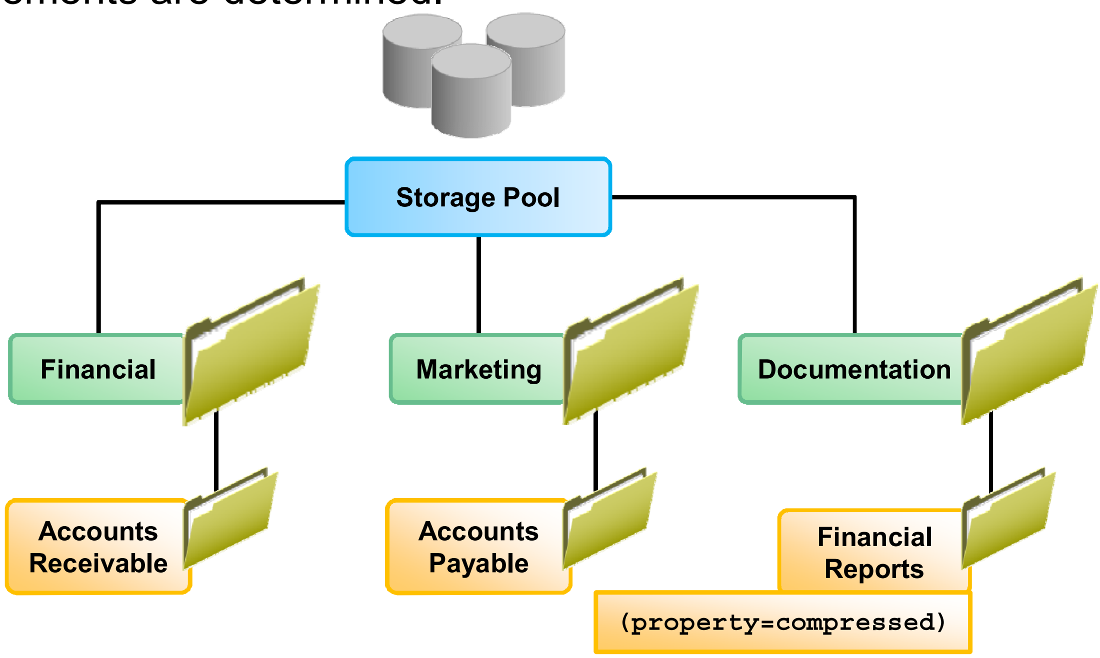
You will find Practice 6-1 and Practice 6-2 in your *Activity Guide*. It should take you about 60 minutes to complete.

Lesson Agenda

- Introducing ZFS
- Administering ZFS Storage Pools
- **Administering ZFS File Systems**
- Administering ZFS Properties
- Administering ZFS Snapshots and Clones

Determining ZFS File System Configuration Requirements

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



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

Determining how the file systems in a pool should be configured is another important part of data storage management planning. First, 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, you should determine the 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.

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 accepts the name of the file system as a single argument to create the file system. In the example in the slide, a file system named `reports` is being created.

Creating a ZFS File System

A file system is mounted:

- Automatically if it is created successfully
- As `/dataset`
- By using the 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` by using the path that 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 in the slide, 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 in the following table:

Condition	Option	Results
File system is busy.	`-f`	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`	Recursively destroys all dependents, including cloned file systems outside the target hierarchy

Caution: No confirmation prompts appear with the `-f`, `-r`, and `-R` options.

ORACLE

If the `zfs destroy` command fails, there are several options that 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 that is described in the preceding paragraph fails. To force the destruction of *all* dependents, including the 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_2014
```

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

You can rename a file system by using the `zfs rename` command. In the example in the slide, you perform a simple renaming of the `reviews` file system that resides in `hrpool/home` to `reviews_2014`.

Renaming a ZFS File System

Example of file system relocation:

```
# zfs rename hrpool/home/jobdesc hrpool/ws/jobdesc
```

- The new location:
  - Must be within the same pool
  - Must have enough space to hold the new file system

The example in the slide 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 renaming, 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 renaming fails.

Renaming a ZFS File System

- The renaming operation attempts an unmount/remount sequence for:
  - The file system
  - Any descendent file systems
- If it is unable to unmount an active file system:
  - The rename operation fails
  - A forced unmount is required

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
...
(output truncated)
```


The `zfs list` command displays the names of all datasets in the system, including their `used`, `available`, `referenced`, and `mountpoint` properties (as illustrated in the example in the slide).

Listing Basic ZFS Information

You can also use `zfs list` to display the following:

- 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 in the slide, basic information for `pool/home/data` is listed by using the `-r` option.

Mounting ZFS File Systems

Use the `zfs mount` command to:

- View ZFS-managed mounted file systems
- Change mount options
- Explicitly mount a file system

Mounting a file system is a process of attaching a file system to the file system hierarchy at *mount_point*, which is the path name of a directory. This helps to make a new created file system available to the system and to users. If *mount_point* has any contents before the mount operation, these are hidden until the file system is unmounted.

With a traditional file system in operating systems other than Oracle Solaris 11, changing a mount point would require:

- Unmounting the file system
- Making a new directory
- Editing `/etc/vfstab`
- Mounting the new file system

With ZFS, this can be done with a single command as you observed in the preceding slides.

By default, all ZFS file systems are mounted by ZFS at boot by using the `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 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 the 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.

Unmounting a file system is the process of detaching a file system from its *mount_point* in the file system directory hierarchy, thereby making it unavailable to the system and to users. 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 in the slide, the file system `hrpool/home/qarpts` is being unmounted by using the file system name.

In the second example, the same file system is being unmounted by using the 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.

In case you forget any of the `zfs` subcommands and their syntaxes, the `zfs` command has a `help` subcommand that you can use to view more information about all its subcommands and their supported options. Example:

```
# zfs help
The following commands are supported:
allow       clone    create     destroy   diff     get
groupspace  help     hold       holds     inherit  list
mount       promote  receive    release   rename   rollback
send        set      share      snapshot  unallow  unmount
unshare     upgrade  userspace
For more info, run: zfs help <command>
# zfs help create
usage:
create [-p] [-o property=value] ... <filesystem>
create [-ps] [-b blocksize] [-o property=value] ... -V <size> <volume>
```

Quiz

Which command is used to create a ZFS file system?

a. `zfs make`
b. `zfs create`
c. `zpool create`
d. `zpool make`

Answer: b

Quiz

Which 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 6-3 Overview: Administering ZFS File Systems

In this practice, you create ZFS file systems in the ZFS storage pools.

This practice should take you about 30 minutes to complete.

Lesson Agenda

- Introducing ZFS
- Administering ZFS Storage Pools
- Administering ZFS File Systems
- **Administering ZFS Properties**
- Administering ZFS Snapshots and Clones

Administering ZFS Properties

- ZFS Properties: Overview
- Types of Native ZFS Properties
- Querying ZFS Properties
- Setting ZFS Properties
- Inheriting ZFS Properties

In this section, you are introduced to the ZFS file system properties, their types, and their uses. You also learn how to query them.

You now take a closer look at each of these topics, beginning with a brief overview of the ZFS file system properties.

ZFS Properties: Overview

- Properties allow you to control the following:
  - File systems
  - Volumes
  - Snapshots
  - Clones
- There are two property types:
  - Native
    - Export internal statistics
    - Control ZFS file system behavior
  - User-defined
    - Have 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. You learn about snapshots and clones in the next section.

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 native properties. If you want to find out more about user properties, refer to http://docs.oracle.com/cd/E36784_01/html/E36835/gdrcw.html#scrolltoc.

Types of Native ZFS Properties

- Read-only statistics
  - Can be retrieved but not set
  - Are not inherited
- Settable
  - Can be both retrieved and set
  - 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

The table in the slide displays a sample of the native ZFS properties, some of which you will encounter in this course and some in the follow-on course *Oracle Solaris 11 Advanced System Administration*.

For a complete list and full descriptions of the native ZFS properties, refer to the http://docs.oracle.com/cd/E36784_01/html/E36835/gazss.html#scrolltoc.

Querying ZFS Properties

You can query property values with:

- `zfs list`
- `zfs get`
  - Complex queries
  - Scripting
  - Any dataset property

The simplest way to query property values is by using the `zfs list` command. However, for complex queries and for scripting, you can use the `zfs get` command to obtain more detailed information in customized format.

Retrieving ZFS Properties

To retrieve any dataset property, use `zfs get`, followed by the property name and the dataset name.

```
# zfs get checksum hrpool/ws
NAME                PROPERTY        VALUE       SOURCE
hrpool/ws           checksum        on          default
```

ORACLE

You can use the `zfs get` command, followed by the property name and the dataset name, to retrieve any dataset property. In the example in the slide, the `zfs get` command is used to retrieve the `checksum` property value for `hrpool/ws`. As you can see, `VALUE` is set to `on` and `SOURCE` is set to `default`.

Retrieving ZFS Properties

The source values in `zfs get` are shown in the following table:

Source Value	Definition
`default`	The property setting was not inherited or set locally.
`local`	The property was explicitly set on the dataset by using the `zfs set` command.
`inherited` from *dataset-name*	The property was inherited from the named ancestor.
`temporary`	This property value was set by using the `zfs mount -o` option, and is valid only for the lifetime of the mount.
`- (none)`	This property is a read-only property. Its value is generated by ZFS.

In addition to the `default`, `local`, and `inherited` from dataset–name source values that you have already seen, you may see two other source values in the `zfs get` output:

- **`temporary`:** Is set by using the `zfs mount -o` option, and is valid only for the lifetime of the mount
- **`- (none)`:** Is a read-only property with a value that is generated by ZFS

Retrieving ZFS Properties

To retrieve all properties for a specified dataset, use `zfs get all`, followed by the dataset name.

```
# zfs get all hrpool
NAME      PROPERTY        VALUE         SOURCE
hrpool    aclinherit      restricted    default
hrpool    aclmode         discard       default
hrpool    atime           on            default
hrpool    available       15.6G         -
hrpool    canmount        on            default
...
<output omitted>
```


You can use the special `all` keyword to retrieve all properties for a specified dataset. The example in the slide uses the `all` keyword to retrieve all existing dataset properties for the `hrpool` file system.

Retrieving ZFS Properties

To specify the property types to display, use `zfs get -s`, followed by the source value and the dataset name.

```
# zfs get -s local all hrpool
NAME      PROPERTY         VALUE     SOURCE
hrpool    compression      on        local
```

With the `-s` option, you can:

- Specify the desired source types with a comma-separated list
- Use the following source types: `default`, `local`, `inherited`, `temporary`, and `none`

You can use the `-s` option with the `zfs get` command to specify the type of properties to display by source value.

This option takes a comma-separated list that indicates the desired source types. Only properties with the specified source type are displayed. The valid source types are `default`, `local`, `inherited`, `temporary`, and `none`.

In the example in the slide, you have specified to show all properties with a `local` source value for the file system `hrpool`. Only the `compression` property met the specified criteria, and has been retrieved.

Retrieving ZFS Properties

The following `zfs get` options are designed for scripting:

- `-H`
  - Omits header information
  - Presents all white space as tabs
- `-o`
  - Allows customization of output
  - Takes a comma-separated list of literal fields to display, together with a separate list of properties

```
# zfs get -H -o value compression hrpool/home
on
```


The `zfs get` command supports the `-H` and `-o` options, which are designed for scripting.

The `-H` option omits any header information and presents all white spaces as tabs.

You can use the `-o` option to customize output. This option takes a comma-separated list of literal fields (`name`, `value`, `property`, and `source`) to be output, followed by a space and an argument, which is a comma-separated list of properties.

The example in the slide shows how to retrieve a single value by using the `-H` and `-o` options with `zfs get`. In the example, you are retrieving the `hrpool/home compression` property value. The value returned is `on`.

This concludes our discussion of setting, inheriting, and querying ZFS properties. Next, you briefly look at mounting and sharing ZFS file systems.

Setting ZFS Properties

To modify any settable dataset property, use `zfs set`, followed by *`property=value`* and the dataset name.

```
# zfs set atime=off hrpool/home
```

Note: Only one property can be set or modified during each `zfs set` invocation.

You can also set a property during the creation of a dataset by using `zfs create`.

```
# zfs create -o atime=off hrpool/home
```


To modify any settable dataset property, use the `zfs set` command, followed by *`property=value`* and a dataset name. In the example in the slide, the `atime` property is set to `off` for `hrpool/home`.

Note: Only one property can be set or modified during each `zfs set` invocation.

You can also use the `zfs create` command to set properties when you are creating the file system. In this example, you are setting the `atime` property to `off` as you create the file system `hrpool/home`.

Now you look at how ZFS property inheritance works.

Inheriting ZFS Properties

- All settable properties inherit their values from their parents.
- All inheritable properties have an associated source.

Source Value	Definition
`default`	The property setting was not inherited or set locally.
`local`	The property was explicitly set on the dataset by using the `zfs set` command.
`inherited` from *dataset-name*	The property was inherited from the named ancestor.

All settable properties, except quotas and reservation properties, inherit their values from their parents.

All inheritable properties have an associated source. The source indicates how a property was obtained. The source of a property can have the following values:

- **`default`:** The property setting was not inherited or set locally. This source is a result of no ancestor having the property as source local.
- **`local`:** The property was explicitly set on the dataset by using the `zfs set` command.
- **`inherited` from *dataset-name***: The property was inherited from the named ancestor.

If no ancestor has an explicit value set for an inherited property, the default value for the property is used.

Inheriting ZFS Properties

```
# zfs list
NAME                          USED  AVAIL  REFER  MOUNTPOINT
datapool                      176K  1.95G    23K  /export/share
datapool/software              65K  1.95G    23K  /export/share/software
datapool/software/solaris      42K  1.95G    21K  /export/share/software/solaris
datapool/software/solaris/ar  21K  1.95G    21K  /export/share/software/solaris/ar
```

```
# zfs get -r compression datapool
NAME                          PROPERTY     VALUE     SOURCE
datapool                      compression  off       default
datapool/software             compression  off       default
datapool/software/solaris     compression  off       default
datapool/software/solaris/ar  compression  off       default
```


You now walk through an example of how ZFS property inheritance works.

- First, you run the `zfs list` command to see the file system hierarchy, as shown in the first output in the slide.
- Next, you run the `zfs get -r` command for the `compression` property, as shown in the second output. Here, you can see that the `compression` property is set to `off`, which is the default for all file systems.

Inheriting ZFS Properties

```
# zfs set compression=on datapool/software/solaris
# zfs get -r compression datapool
NAME                          PROPERTY     VALUE   SOURCE
datapool                      compression  off     default
datapool/software             compression  off     default
datapool/software/solaris     compression  on      local
datapool/software/solaris/ar  compression  on      inherited from datapool/software/solaris
```


Next, you set the `compression` property to `on` for the `datapool/software/solaris` file system.

If you run the `zfs get -r compression` command for `datapool`, you see how the child file system inherits the ZFS property. Note that the compression value for `datapool/software/solaris` is now set to `on` and the `SOURCE` has changed from `default` to `local`.

You also see how `datapool/software/solaris/ar` has inherited the `compression` property value of `on` from its parent file system.

Inheriting ZFS Properties

To clear a property setting and have the setting inherited from the parent, use `zfs inherit`, followed by the property name and the system file name path.

```
# zfs inherit compression datapool/software/solaris
# zfs get -r compression datapool
NAME                          PROPERTY     VALUE     SOURCE
datapool                      compression  off       default
datapool/software             compression  off       default
datapool/software/solaris     compression  off       default
datapool/software/solaris/ar  compression  off       default
```


If you want to return the `datapool/software/solaris` file system's compression value to `off`, you can do so by using the `zfs inherit` command. This command clears a property setting, thereby causing the setting to be inherited from the parent.

In the example in the slide, `zfs inherit` is used to unset the `compression` property, thereby causing the property to inherit the default setting of `off`. Because neither `datapool` nor `datapool/software` has the `compression` property set locally, the default value is used. If both `datapool` and `datapool/software` have `compression` `on`, the value set in the most immediate ancestor is used.

Now you look at how to query ZFS properties.

Lesson Agenda

- Introducing ZFS
- Administering ZFS Storage Pools
- Administering ZFS File Systems
- Administering ZFS Properties
- **Administering ZFS Snapshots and Clones**

Administering ZFS Snapshots and Clones

- Administering ZFS Snapshots
  - Creating a ZFS Snapshot
  - Displaying a ZFS Snapshot
  - Renaming a ZFS Snapshot
  - Holding a ZFS Snapshot
  - Rolling Back a ZFS Snapshot
  - Destroying a ZFS Snapshot
  - Snapshot Space Accounting
  - Identifying ZFS Snapshot Differences
- Administering ZFS Clones
  - Creating a ZFS Clone
  - Relationship of Clone and Snapshot
  - Replacing a ZFS File System with a ZFS Clone
  - Destroying a ZFS Clone

In this section, you learn how to administer ZFS snapshots and clones.

ZFS Snapshots

- Are a read-only view of a file system or volume
- Can be created quickly and easily
  - Unlimited number of snapshots are 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, the consumed space is released.

To protect data that is stored in ZFS storage pools and on 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. Initially, snapshots consume no additional space within the pool because they provide pointers to 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 snapshots include the following features:

- They persist across system reboots.
- The theoretical maximum number of snapshots is 2^{64}.
- Snapshots use no separate backing store. They consume disk space directly from the same storage pool as the file system or volume from which they were created.
- Recursive snapshots are created quickly as one atomic operation. The snapshots are created together (all at once) or not created at all. The benefit of atomic snapshot operations is that the snapshot data is always taken at one consistent time, even across descendent file systems.

Snapshots of volumes cannot be accessed directly, but they can be cloned, backed up, rolled back, and so on.

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` accepts the name of the snapshot as its only argument to create a snapshot.

In the example in the slide, 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 in the example in the slide, 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 take a closer look at this option when you learn about displaying and accessing snapshots.

Now, you look at how to destroy a ZFS snapshot.

Note: The `zfs snapshot` command has a `snap` alias that provides abbreviated syntax for this command. Example:

```
# zfs snap -r hrpool/home@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
- Is disabled by default
- Is enabled by using `zpool set listsnapshot=on` *`<poolname>`*

You use the `zfs list -t snapshot` command to display snapshots as shown in the example in the slide.

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 the 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       Thur Nov 28 10:08 2013
hrpool/home/reports@wednesday     Fri  Nov 30 08:05 2013
hrpool/home/reports@thursday      Mon  Dec  2 07:03 2013
hrpool/home/bonus@now             Tue  Dec  3 06:15 2013
```

ORACLE

You can list the 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 in the slide, 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 that you can use with the `zfs list -t snapshot` subcommand, see the `zfs`(1) man page.

Renaming a ZFS Snapshot

To rename a snapshot, use `zfs rename` followed by the snapshot name.

```
# zfs rename hrpool/home/report@121014 hrpool/home/report@today
```

Note: Snapshots must be renamed within the same pool and dataset from which they were created.

You can rename a snapshot by using the `zfs rename` command followed by the snapshot name. In the example in the slide, the snapshot `report@121014` that resides in `hrpool/home` is renamed to `report@today`.

Note: Snapshots must be renamed within the same pool and dataset from which they were created.

Renaming a ZFS Snapshot

To recursively rename snapshots, use `zfs rename -r` followed by the snapshot name.

```
# zfs list
NAME                          USED    AVAIL  REFER    MOUNTPOINT
users                         270K    16.5G   22K     /users
users/home                    76K     16.5G   22K     /users/home
users/home@yesterday          0       -       22K     -
users/home/jjones             18K     16.5G   18K     /users/home/jjones
users/home/jjones@yesterday   0       -       18K     -
# zfs rename -r users/home@yesterday @2daysago
# zfs list -r users/home
NAME                          USED    AVAIL  REFER    MOUNTPOINT
users/home                    76K     16.5G   22K     /users/home
users/home@2daysago           0       -       22K     -
users/home/jjones             18K     16.5G   18K     /users/home/jjones
users/home/jjones@2daysago    0       -       18K     -
```


You can recursively rename snapshots with the `zfs rename -r` command followed by the snapshot name.

In the example in the slide, only those snapshots that are named `@yesterday` are renamed to `@2daysago`.

You can see a before-and-after picture with the list of file systems before the `zfs rename -r` command is run in the first half of the example, and then the change that has occurred in the second half of the example. In short, any snapshot with the `@yesterday` name was changed to `@2daysago` after the `rename` command was run.

Next, you look at how to display and access ZFS snapshots.

Holding a ZFS Snapshot

- The snapshot hold feature:
  - Prevents a snapshot from being destroyed by using `zfs destroy`
  - Allows a snapshot with clones to be deleted pending the removal of the last clone by using `zfs destroy -d`
- The snapshot user-reference count:
  - Is initialized to zero
  - Increases by one whenever a hold is put on the snapshot
  - Decreases by one whenever a hold is released
  - Must be at zero before the snapshot can be destroyed

Remote replication of ZFS datasets can result in different automatic snapshot policies on the two sides of a replication pair. For example, the sending side may want to keep five snapshots at one-minute intervals, whereas the receiving side may want to keep 10 snapshots at one-minute intervals. This can result in the older snapshots being destroyed inadvertently by `zfs receive` because they no longer exist on the sending side.

The ZFS snapshot hold feature addresses this issue. Holding a snapshot (`zfs hold`) prevents it from being destroyed. If a hold exists on a snapshot, you will not be able to destroy it by using the `zfs destroy` command. You look at the two options that you have for destroying a held snapshot in subsequent slides.

In addition, the snapshot hold feature allows a snapshot with clones to be deleted pending the removal of the last clone by using the `zfs destroy -d` command. You take a closer look at how this is done in the subsequent slides.

Each snapshot has an associated user-reference count, which is initialized to zero. This count increases by one whenever there is a hold on the snapshot and decreases by one whenever the hold is released. As discussed, a snapshot can be destroyed only if it has no clones. In Oracle Solaris 11, the snapshot must also have a user-reference count of zero before it can be destroyed.

Holding a ZFS Snapshot

To hold a snapshot or a set of snapshots, use `zfs hold keep` followed by the snapshot name.

```
# zfs hold keep hrpool/home/report@snap1
```

To recursively hold the snapshots of all descendant file systems, use `zfs hold` with `-r`, followed by `keep` and the snapshot name.

```
# zfs hold -r keep hrpool/home@now
```

Note: Each snapshot has its own tag namespace, and tags must be unique within that space. `keep` is only a tag.

To hold a snapshot or a set of snapshots, use the `zfs hold keep` command followed by the snapshot name. In the first example in the slide, a hold tag (`keep`) is put on `hrpool/home/report@snap1`.

You can use the `-r` option with the `zfs hold` command and the `keep` hold tag to recursively hold the snapshots of all descendant file systems, as shown in the second example. Here, you are holding the snapshots of all the descendant file systems of `hrpool/home@now`.

Note: Each snapshot has its own tag namespace, and tags must be unique within that space. In this example, `keep` is a user-defined tag.

Holding a ZFS Snapshot

To display a list of held snapshots, use `zfs holds` followed by the snapshot name.

```
# zfs holds hrpool/home@now
NAME             TAG   TIMESTAMP
hrpool/home@now  keep  Mon Mar 10 12:40:12 2014
```

To display a recursive list of held snapshots, use `zfs holds` with `-r`, followed by the snapshot name.

```
# zfs holds -r hrpool/home@now
NAME                     TAG   TIMESTAMP
hrpool/home/report@now   keep  Mon Mar 10 12:40:12 2014
hrpool/home/jjones@now   keep  Mon Mar 10 12:40:12 2014
hrpool/home@now          keep  Mon Mar 10 12:40:12 2014
```


You can use the `zfs holds` command followed by the snapshot name to display a list of held snapshots. In the first example in the slide, snapshot holds are displayed for `hrpool/home@now`. Notice that the output returns the name of the snapshot, the tag name (in this case, `keep`), and the time stamp.

You can use the `-r` option with the `zfs holds` command and the snapshot name to get a recursive list, as illustrated in the second example.

Holding a ZFS Snapshot

Two options to destroy a held ZFS snapshot:

1. Use `zfs destroy –d` followed by the snapshot name, and then release the snapshot hold, which removes the snapshot.
2. Release the held snapshot, and then destroy it by using `zfs destroy`.

To release a held snapshot or a set of snapshots, use `zfs release -r` followed by `keep` and the snapshot name.

```
# zfs release -r keep hrpool/home@now
```

Note: `-r` enables a recursive release of the hold.

As mentioned before, if a hold exists on a snapshot, you will not be able to destroy it by using the `zfs destroy` command. To destroy the snapshot, you have two options:

- You can destroy the held snapshot by using the `zfs destroy -d` command followed by the snapshot name, and then release the snapshot hold, which removes the snapshot.
- You can release the snapshot, and then destroy it by using the `zfs destroy` command without the `-d` option.

Note: If a held snapshot has associated clones, you must destroy the clones first before you can destroy the held snapshot.

To release a hold on a snapshot or a set of snapshots, use the `zfs release` command with the `-r` option, followed by the hold tag `keep` and the snapshot name. `-r` enables a recursive release of the hold, and is optional. In the example in the slide, you are releasing the recursive hold on the `hrpool/home@now` snapshot.

This snapshot can be destroyed if all the holds have been released.

Holding a ZFS Snapshot

Snapshot hold properties:

- **`defer_destroy`:** Set to `on` if the snapshot is marked for deferred destruction by using the `zfs destroy -d` command
- **`Userrefs`:** Set to the number of holds on the snapshot

The snapshot hold information is identified through two properties:

- The `defer_destroy` property is `on` if the snapshot is marked for deferred destruction by using the `zfs destroy -d` command. Otherwise, the property is `off`.
- The `userrefs` property is set to the number of holds on this snapshot, which is also referred to as the *user-reference count*.

Holding a ZFS Snapshot

To view the ZFS snapshot hold properties, use `zfs get -r defer_destroy,userrefs` followed by the file system name.

```
# zfs get -r defer_destroy,userrefs hrpool/home
NAME                      PROPERTY       VALUE    SOURCE
hrpool                    defer_destroy  -        -
hrpool                    userrefs       -        -
hrpool/home               defer_destroy  -        -
hrpool/home               userrefs       -        -
hrpool/home/report@now    defer_destroy  off      -
hrpool/home/report@now    userrefs       1        -
hrpool/home/jjones@now    defer_destroy  off      -
hrpool/home/jjones@now    userrefs       1        -
hrpool/home@now           defer_destroy  off      -
hrpool/home@now           userrefs       1        -
```


You can view these properties by using the `zfs get -r` command followed by the comma-separated property name and the file system name.

In the example in the slide, the `defer_destroy` and `userrefs` properties for `hrpool/home` are displayed. As you can see from the output, each of the `@now` snapshots has the `defer_destroy` property set to `off`, which is the default, and a value of `1` for the `userrefs` property, which indicates that each of these snapshots has one hold on it.

Rolling Back a ZFS Snapshot

To discard all the changes made since a specific snapshot, enter `zfs rollback` followed by the snapshot name.

```
# zfs rollback hrpool/home/qarpt@thursday
```

By default, `zfs rollback` rolls back only to the most recent snapshot.

To destroy more recent snapshots, enter `zfs rollback` with `-r`, followed by the snapshot name.

```
# zfs rollback -r hrpool/home/qarpt@tuesday
```

ORACLE

The `zfs rollback` command causes the file system to revert to its state at the time the snapshot was taken. By default, the `zfs rollback` command cannot roll back to a snapshot other than the most recent snapshot.

You can use the `zfs rollback` command followed by the snapshot name to discard all the changes made since a specific snapshot. In the example in the slide, the `hrpool/home/qarpt` file system is rolled back to the `thursday` snapshot.

To roll back to an earlier snapshot, you must destroy all intermediate snapshots. To do this, you must specify the `-r` option with the `zfs rollback` command followed by the snapshot name, as shown in the second example. Here, the `hrpool/home/qarpt` file system is rolled back to the `tuesday` snapshot. For this operation to take place, the `wednesday` and `thursday` snapshots must be destroyed.

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 in the slide, you are destroying the snapshot named `hrpool/home/reports@now`.

Snapshot Space Accounting

- When a snapshot is created, its space:
  – Is initially shared between the snapshot and the file system
  – Is possibly shared with previous snapshots
- As the file system changes, the 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 that of the file system when the snapshot was created.

Snapshot Space Accounting

To display how much space is consumed by snapshots and descendant file systems, use `zfs list -o space`.

```
$ zfs list -o space
NAME                             AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
rpool                            2.20G  13.2G         0   4.97M              0      13.2G
rpool/ROOT                       2.20G  4.49G         0     31K              0      4.49G
rpool/ROOT/solaris               2.20G  4.49G     70.2M   4.08G              0       354M
rpool/ROOT/solaris/var           2.20G   354M      156M    198M              0          0
rpool/VARSHARE                   2.20G   156K         0     93K              0        63K
rpool/VARSHARE/pkg               2.20G    63K         0     32K              0        31K
rpool/VARSHARE/pkg/repositories  2.20G    31K         0     31K              0          0
rpool/dump                       2.22G   792M         0    768M          24.4M          0
rpool/export                     2.20G  6.88G         0     34K              0      6.88G
rpool/export/IPS                 2.20G  6.77G         0   6.77G              0          0
rpool/export/home                2.20G   108M         0     32K              0       108M
rpool/export/home/oracle         2.20G   108M         0    108M              0          0
rpool/swap                       2.23G  1.03G         0   1.00G          32.5M          0
```


From the output shown in the example in the slide, you can see the space that is available in each file system, the space that is being used, the space that is consumed by snapshots of each dataset (`USEDSNAP`), the space that is used by the dataset itself (`USEDDS`), the space that is used by a refreservation set on the dataset (`USED REFRESERV`), and the space that is used by the children of this dataset (`USEDCHILD`).

Identifying ZFS Snapshot Differences

To determine ZFS snapshot differences, use `zfs diff` followed by the snapshot names.

```
# zfs snapshot datapool/hrdata@before
# touch /datapool/hrdata/newfile
# zfs snapshot datapool/hrdata@after
# zfs list -r -t snapshot -o name,creation
NAME                                CREATION
datapool/hrdata@before              Thu Oct 24 14:54 2013
datapool/hrdata@after               Thu Oct 24 14:59 2013
rpool/ROOT/solaris@install          Tue Oct 24 22:33 2013
# zfs diff datapool/hrdata@before datapool/hrdata@after
M/datapool/hrdata/
+/datapool/hrdata/newfile
#
```


To determine the differences between ZFS snapshots, you can use the `zfs diff` command. The output of this command provides a high-level description of the differences between a snapshot and a descendant dataset. The descendant can be either a snapshot of the dataset or the current dataset. For each file that has undergone a change between the original snapshot and the descendant, the type of change is described along with the name of the file. In the case of a rename, both the old and new names are shown. The type of change follows any time stamp displayed, and is described with a single character. The definition of each of these characters is provided in the next slide.

In the example in the slide:

1. A `before` snapshot of the `datapool/hrdata` ZFS file system was taken
2. A new file (`newfile`) was created in `/datapool/hrdata`
3. Another snapshot (`after`) of the same ZFS file system was taken
4. The `zfs list` command is used to list the snapshots based on name and creation date
5. The `zfs diff` command is run to determine the differences between the `before` and `after` snapshots

The `M` in the `zfs diff` command output indicates that the `/datapool/hrdata/` directory has been modified, and the `+` indicates that a file `/datapool/hrdata/newfile` exists in the later snapshot.

The `zfs diff` command provides an enumeration option, `-e`, to identify all the files that were added or modified between the two snapshots. The generated output identifies all files added, but does not provide possible deletions. Example:

```
# zfs diff -e datapool/hrdata@before datapool/hrdata@after
+     /datapool/hrdata/
+     /datapool/hrdata/newfile
```

You can also use the `-o` option to select fields to be displayed. Example:

```
# zfs diff -e -o size -o name datapool/hrdata@before \
datapool/hrdata@after
+     3       /datapool/hrdata/
+     209655  /datapool/hrdata/newfile
```

Identifying ZFS Snapshot Differences

File or Directory Change	Identifier
File or directory is modified, or file or directory link has changed.	`M`
File or directory is present in the older snapshot but not in the newer snapshot.	`-`
File or directory is present in the newer snapshot but not in the older snapshot.	`+`
File or directory is renamed.	`R`

The table in this slide summarizes the file or directory changes that are identified by the `zfs diff` command. For more information about the `zfs diff` command, see `zfs(1M)`.

Now that you have an idea of how to work with ZFS snapshots, you turn your attention to ZFS clones.

ZFS Clones

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

A snapshot is a read-only point-in-time copy of a file system, and 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. The `zfs clone` command is used to specify the snapshot from which to create the clone.

Clones can be created only from a snapshot. A clone is related to the snapshot from which it originated. When a snapshot is cloned, an implicit dependency is created between the clone and the snapshot. After a clone is created, the snapshot from which it originated cannot be deleted unless the clone is deleted first. The `origin` property exposes this dependency, and the `zfs destroy` command lists any such dependencies, if they exist.

Clones do not inherit the properties of the dataset from which it was created. Use the `zfs get` and `zfs set` commands to view and change the properties of a cloned dataset.

Clones provide an extremely space-efficient way to store many copies of mostly shared data such as workspaces, software installations, and diskless clients.

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.

In the example in the slide, create a clone named `hrpool/home/reports/bug123` that has 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 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 the 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 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.

Replacing a ZFS File System with a ZFS Clone

With the clone replacement process, you can:

- Clone and replace file systems so that the original file system becomes the clone of the newly created file system
- Destroy the file system from which the clone was originally created

Note: Without clone promotion, you cannot destroy the original file system of active clones.

ZFS uses a process called *clone replacement* (or *promotion*), which enables you to replace an active ZFS file system with a clone of that file system. This process facilitates the ability to clone and replace file systems so that the original file system becomes the clone of the newly created file system. In addition, this process makes it possible to destroy the file system from which the clone was originally created.

Note: Without clone promotion, you cannot destroy the original file system of active clones.

This feature is good to have if, for some reason, you need to modify production data. (In normal circumstances, you would not use this method. You would introduce the modifications through databases and other tools instead.) To accommodate this special situation, create a clone from the file system's snapshot. This clone is nothing but a copy of the production file system; however, it is writeable and editable. So now a programmer (or system administrator) can open it like any other file system, modify the data and, considering it as a newer copy of the file system, overlay the production file system with it. Because you do not want to modify production data directly, being able to use clones is useful.

Replacing a ZFS File System with a ZFS Clone

To replace an active ZFS file system with a clone of that file system, use `zfs promote` followed by the clone name.

```
# zfs snapshot hrpool/reviews/q4@today
# zfs clone hrpool/reviews/q4@today hrpool/reviews/q4sum
# zfs list -r hrpool/reviews
NAME                       USED   AVAIL  REFER  MOUNTPOINT
hrpool/reviews             314K   8.24G  25.5K  /hrpool/reviews
hrpool/reviews/q4          288K   8.24G  288K   /hrpool/reviews/q4
hrpool/reviews/q4@today       0   -      288K   -
hrpool/reviews/q4sum          0   8.24G  288K   /hrpool/reviews/q4sum
```

```
# zfs promote hrpool/reviews/q4sum
# zfs list -r hrpool/reviews
NAME                       USED   AVAIL  REFER  MOUNTPOINT
hrpool/reviews             316K   8.24G  27.5K  /hrpool/reviews
hrpool/reviews/q4             0   8.24G  288K   /hrpool/reviews/q4
hrpool/reviews/q4sum       288K   8.24G  288K   /hrpool/reviews/q4sum
hrpool/reviews/q4@today       0       -  288K   -
```


The clone replacement process is a two-step operation.

In the first slide example, you see from the `zfs list` output that the `hrpool/reviews/q4` file system has been cloned by using the `hrpool/review/q4@today` snapshot.

In the second example, the original `hrpool/reviews/q4` file system is replaced with the cloned file system, `hrpool/reviews/q4sum`, by promoting the clone.

In the `zfs list` output, you can see that the storage used by the original `q4` file system has been replaced with the `q4sum` file system.

Replacing a ZFS File System with a ZFS Clone

To rename the promoted file systems to the original name, use `zfs rename` followed by the current file system name and a new file system name.

```
# zfs rename hrpool/reviews/q4 hrpool/reviews/q4legacy
# zfs rename hrpool/reviews/q4sum hrpool/reviews/q4
# zfs list -r hrpool/reviews
NAME                          USED  AVAIL   REFER   MOUNTPOINT
hrpool/reviews                316K  8.24G   27.5K   /hrpool/reviews
hrpool/reviews/q4             288K  8.24G   288K    /hrpool/reviews/q4
hrpool/reviews/q4@today          0      -   288K    -
hrpool/reviews/q4legacy          0 8.24G    288K    /hrpool/reviews/q4legacy
```


After you have replaced the original ZFS file system with the clone, you will want to complete the second and final step of the process by renaming the file systems.

To do this, use the `zfs rename` command followed by the current file system name and a new file system name.

In the example in the slide, you renamed the original `hrpool/reviews/q4` file system to `hrpool/reviews/q4legacy` and the original clone file system, `hrpool/reviews/q4sum`, to `hrpool/reviews/q4`, which was the name of the original file system.

In the `zfs list` output, you can see that the name changes have taken effect.

The original file system and snapshot can now be deleted.

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 in the slide, 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

Practice 6-4 Overview: Administering ZFS Snapshots and Clones

This practice covers the following tasks:

- Creating ZFS snapshots
- Creating ZFS clones

This practice should take you approximately 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Describe ZFS and its features
- Administer ZFS storage pools
- Administer ZFS file systems
- Administer ZFS properties
- Administer ZFS snapshots and clones