

D80397GC10

Edition 1.0

September 2013

D83665

ORACLE®

Oracle Solaris 11 Fault Analysis and Troubleshooting

Student Guide

Authors

Kristi Herd
Tammy Shannon

**Technical Contributors
& Reviewers**

Art Beckman
Harry Burks
Dimitri De Wild
Prashant Deo
Rex di Bona
Glynn Foster
John Hathaway
Rosemary Martinak
Jeff McMeekin
Kristi McNeill
Fermina Quinones
Gary Riseborough
Isaac Rozenfeld
Michael Stapleton
Martha Starkey
Todd Lowry

Editors

Smita Kommini
Raj Kumar
Anwesha Ray

Graphic Designer

Divya Thallap

Publishers

Pavithran Adka
Nita Brozowski
Sujatha Nagendra
Michael Sebastian

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

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

Contents

Preface

1 Introduction

- Overview 1-2
- Course Goals 1-3
- Course Agenda: Day 1 1-4
- Course Agenda: Day 2 1-6
- Course Agenda: Day 3 1-8
- Course Agenda: Days 4 and 5 1-9
- Introductions 1-10
- Your Learning Center 1-11
- Your Practice Environment 1-12
- Accessing Your Practice Environment 1-13

2 Analyzing System Failures and Troubleshooting Techniques

- Course Map 2-2
- Objectives 2-3
- Agenda 2-4
- Identifying the Basic Layers of an Oracle Solaris System 2-5
- Identifying Error Types 2-6
- Two-Phased Approach to Troubleshooting 2-9
- Agenda 2-10
- High-Level Troubleshooting Strategy 2-11
- Agenda 2-12
- Clarifying the Problem Statement 2-13
- Interpreting System Messages 2-14
- Gathering and Interpreting Log Files 2-16
- Using Debug and Verbose Mode 2-17
- Log File: Example 2-18
- System Panics and Crash Dumps 2-19
- Displaying the Current Crash Dump Configuration 2-20
- Modifying the Crash Dump Configuration 2-21
- Saving the Crash Dump File 2-23
- Uncompressing the Crash Dump File 2-24
- Analyzing the Crash Dump File Contents 2-25

- mdb Essentials 2-26
- Forcing a Crash Dump from OpenBoot PROM 2-28
- Forcing a Crash Dump on x86 (or SPARC): Example 2-29
- Using kmdb to Force a Crash Dump 2-30
- Process Core Dumps 2-31
- Managing Process Core Dumps 2-32
- Manually Forcing Process Core Dumps 2-34
- Using gcore with pstack 2-35
- Researching Previously Identified Problems 2-36
- Agenda 2-37
- Mitigating Fault Impact 2-38
- Using SMF, FMA, SNMP, SMTP, and Auto Service Request (ASR) Notifications 2-39
- Running Oracle Explorer or Other Data-Collecting Scripts 2-40
- FMA Features 2-42
- Interpreting Notifications of Faults and Defects 2-44
- Gathering and Interpreting FMA Data by Using fmadm, fmstat, and fmdump 2-45
- Gathering and Interpreting FMA Data by Using fmadm: Example 2-47
- Gathering and Interpreting FMA Data by Using fmstat: Example 2-48
- Gathering and Interpreting FMA Data by Using fmdump: Example 2-49
- Using ZFS Snapshots or Alternate Boot Environments (BEs) as a Safeguard 2-50
- File System Backup and Recovery with Snapshots 2-51
- Using Snapshots for Backup and Recovery 2-53
- Using Alternate Boot Environments (BEs) 2-54
- Alternate Boot Environments: Example 2-55
- Agenda 2-56
- Case Study 1: System Panic 2-57
- Case Study 1: Troubleshooting Steps 2-58
- Case Study 1: Walkthrough 2-61
- Practice 2 Overview 2-67
- Summary 2-68

3 Troubleshooting System Hardware Faults

- Course Map 3-2
- Objectives 3-3
- Agenda 3-4
- Oracle Enterprise-Class Servers 3-5
- System Block Diagram 3-7
- Agenda 3-8
- Hardware Fault Behavior 3-9
- Diagnosing Hardware Faults 3-10

Hardware Troubleshooting Flowchart	3-11
Agenda	3-12
Service Processor (SP)	3-13
Integrated Lights Out Manager (ILOM)	3-14
What Can You Do with ILOM?	3-15
ILOM Fault Handling	3-16
Entering and Exiting ILOM	3-17
ILOM Troubleshooting Tasks	3-18
ILOM Fault Detection: Examples	3-20
Automatic System Recovery (ASR)	3-21
Agenda	3-22
Boot PROM for SPARC Systems	3-23
Boot PROM features	3-24
Boot PROM Initialization Sequence	3-26
Checking the OBP Firmware Revision	3-28
Identifying the OBP Variables Used in Troubleshooting	3-29
Displaying OBP Variables	3-30
Using the setenv and set-default Commands	3-31
Using the eeprom Command	3-32
Using Manual OBP Diagnostic Commands	3-33
Using the probe Commands	3-34
Using the test Commands	3-35
Using the watch Commands	3-36
Quiz	3-37
Agenda	3-39
Introducing POST	3-40
Identifying POST Components	3-41
Enabling Extended POST Diagnostics	3-42
Setting the diag-switch? Variable	3-43
Setting the diag-level Variable	3-44
Using the show-post-results Command	3-45
OBP Device Tree: Overview	3-46
Examining Device Path Naming	3-48
Solaris path_to_inst File	3-49
Listing Device Tree Information	3-50
Using the show-devs Command	3-51
Using the show-disks Command	3-52
Using the show-nets Command	3-53
Booting from the diag-device or boot-device Variable	3-54
Using the diag-device Variable	3-55
Using the boot-device Variable	3-56

- Quiz 3-57
- Agenda 3-60
- Using the prtdiag Command 3-61
- Oracle VTS 3-62
- VTS Testing Modes 3-63
- Installing Oracle VTS 3-65
- VTS Troubleshooting Tasks 3-66
- VTS Logs 3-67
- VTS Messages 3-68
- Agenda 3-69
- Case Study 2: Hardware Failure 3-70
- Case Study 2: Troubleshooting Steps 3-71
- Case Study 2: Walkthrough 3-72
- Practice 3 Overview 3-76
- Summary 3-77

4 Troubleshooting Solaris Startup Faults

- Course Map 4-2
- Objectives 4-3
- Agenda 4-4
- Boot PROM Initialization Sequence 4-5
- Boot Process 4-7
- How Oracle Solaris Boot Archives Are Managed 4-9
- Agenda 4-11
- Booting in Verbose Mode 4-12
- Booting from an Incorrect Device or Disk 4-13
- Booting from an Incorrect Device Type 4-14
- Booting from an Incorrect Disk 4-15
- Recovering a Corrupted Boot Block 4-16
- Booting a System with the Kernel Debugger (kmdb) Enabled 4-17
- Agenda 4-18
- SMF Architecture and Features Review 4-19
- Monitoring SMF Services 4-20
- Determining System Readiness Using Milestones 4-21
- Identifying Service States 4-22
- Setting Up Service State Transition Notifications 4-23
- Installing the smtp-notify Package 4-24
- Enabling the smtp-notify:default Service 4-25
- Configuring Service State Transition Notifications 4-26
- Service State Transition Notification Example 4-27
- Managing Service State Transition Notifications 4-29

Managing SMF Services	4-30
SMF Configuration Repository	4-31
Exploring the SMF Layers	4-32
Service Bundles: Manifests and Profiles	4-34
Viewing SMF Repository Information	4-35
Quiz	4-36
Agenda	4-37
Debugging a Service That Is Not Starting	4-38
Interpreting SMF Log Files and Trees	4-39
Interpreting SMF Log Files and Trees Example	4-40
Troubleshooting Service Dependencies	4-41
Troubleshooting Service Methods	4-43
Using SMF Repository Backups	4-44
SMF Repository Integrity Check Process	4-45
Repairing a Corrupt Repository Example	4-46
Using SMF Repository Snapshots	4-48
Reverting to an SMF Snapshot	4-49
Reverting to an SMF Snapshot Example	4-50
Observing the SMF Portion of the Boot Process	4-51
Adding Truss to a Service	4-52
Quiz	4-53
Agenda	4-54
Solaris Boot Environments (BEs)	4-55
Alternate Boot Environments Example	4-56
Booting a Backup Boot Environment	4-57
Case Study 3: Solaris Fails to Start!	4-58
Case Study 3: Troubleshooting Steps	4-59
Case Study 3: Walkthrough	4-60
Practice 4: Overview	4-66
Summary	4-67

5 Troubleshooting Image Packaging System (IPS) Problems

Course Map	5-2
Objectives	5-3
Agenda	5-4
Reviewing the IPS Features	5-5
Reviewing How IPS Works	5-6
Agenda	5-7
Example Problem	5-8
Determining the Client Host and Domain Names	5-9
Checking Network Connectivity	5-10

- Verifying the Local IPS Publisher Setting 5-11
- Testing Client Access to the Local IPS Server 5-12
- Disabling and Enabling a Publisher 5-13
- Changing a Publisher Origin URI 5-14
- Agenda 5-15
- Previewing a Package Installation Operation 5-16
- Verifying Package Installation 5-17
- Rejecting a Package 5-18
- Using the Group Dependency and Avoid List 5-19
- Requesting the Newest Version of a Package 5-20
- Improved Error Message Example #1 5-21
- Improved Error Message Example #2 5-22
- Improved Error Message Example #3 5-23
- Backing Out Packages 5-24
- For More Information 5-25
- Quiz 5-26
- Agenda 5-28
- Checking the History Log 5-29
- Determining the Package Payload 5-32
- Determining Package Actions and Dependencies 5-33
- Verifying Package Installation 5-34
- Fixing Verification Errors 5-35
- Restoring a File in a Package 5-36
- Quiz 5-37
- Agenda 5-38
- Case Study 4: DTrace Not Working Properly! 5-39
- Case Study 4: Troubleshooting Steps 5-40
- Case Study 4: Walkthrough 5-41
- Additional Resources 5-46
- Practice 5: Overview 5-47
- Summary 5-48

6 Troubleshooting Automated Installer (AI) Problems

- Course Map 6-2
- Objectives 6-3
- Agenda 6-4
- Automated Installation Overview 6-5
- Reviewing the Automated Installation Process 6-6
- Reviewing How AI Works 6-7
- AI Components 6-8
- AI Manifest Excerpts 6-9

AI Client Criteria Examples	6-10
AI SC Profile Excerpts	6-11
Agenda	6-12
Checking Logs and Other Files	6-13
Gathering AI Client Logs and Other Files	6-14
Gathering AI Server Logs and Other Files	6-15
Using Additional Debugging Tools	6-16
Checking DNS	6-17
SPARC Network Booting Errors and Possible Causes	6-18
SPARC Error Messages	6-20
AI Client Boot Failure	6-22
Verifying the Local DHCP Configuration	6-23
Deploying the OS Using AI	6-24
Deploying Non-Global Zones Using AI	6-25
Common AI Problems	6-26
Agenda	6-27
Case Study 5: AI Fails to Deploy Solaris OS!	6-28
Case Study 5: Troubleshooting Steps	6-29
Case Study 5: Walkthrough	6-30
More Information	6-42
Additional Resources	6-43
Quiz	6-44
Practice 6: Overview	6-46
Summary	6-47

7 Troubleshooting Problems with Zones

Course Map	7-2
Objectives	7-3
Agenda	7-4
Global and Non-Global Zone Features and Primary Functions	7-5
Reviewing Immutable Zones Technology	7-6
Reviewing Non-Global Zone States	7-8
Reviewing How Zones Work: Global Zone	7-10
Reviewing How Zones Work: Non-Global	7-11
Reviewing How Zones Work: Example	7-12
Reviewing How Zones Work with AI, IPS, and SMF	7-13
Reviewing How AI Installs Non-Global Zones	7-14
Reviewing How to Specify Non-Global Zones in the Global Zone AI Manifest	7-15
Troubleshooting Non-Global Zones Installation Errors	7-16
Reviewing How Zones Work with IPS Commands, Package Publishers, and System Repository	7-17

Reviewing How Zones Work with IPS Software Package Installation and Updates	7-18
Reviewing How Zones Work with Physical Networks	7-20
Reviewing How Zones Work with Virtual Networks	7-21
Quiz	7-22
Agenda	7-23
Checking the Zones Configuration	7-24
Displaying the Current Zones Configuration on the System	7-25
Determining the Current Zone Configuration	7-26
Displaying a Zone Configuration	7-27
Displaying Zone Network Information	7-29
Running DTrace in a Non-Global Zone	7-30
Running DTrace in a Non-Global Zone: Example	7-31
Shutting Down and Starting Up a Non-Global Zone	7-32
Halting a Non-Global Zone	7-33
Troubleshooting Zones That Do Not Halt	7-34
Uninstalling a Non-Global Zone	7-35
Troubleshooting Zones That Do Not Uninstall	7-36
Cloning a Non-Global Zone	7-39
Cloning a Non-Global Zone Error	7-40
Quiz	7-41
Agenda	7-42
Identifying the Zones-Related SMF Services and Their Roles	7-43
Using SMF Troubleshooting Techniques Within a Non-Global Zone	7-45
Quiz	7-46
Agenda	7-47
Using zonestat	7-48
Using fsstat	7-50
Quiz	7-51
Agenda	7-52
Analyzing Package Version Problems Between a Non-Global Zone and the Global Zone	7-53
Verifying the Zones Services Are Online	7-55
Verifying Communication Between the Zones and the Publisher: Example	7-56
Choosing Different Origins That Are Configured in the Global Zone	7-57
Checking the Zone State	7-59
Agenda	7-60
Case Study 6: Zone Fails to Attach!	7-61
Case Study 6: Troubleshooting Steps	7-62
Case Study 6: Walkthrough	7-63
Additional Resources	7-70

Practice 7: Overview 7-71

Summary 7-72

8 Troubleshooting Physical and Virtual Network Problems

Course Map 8-2

Objectives 8-3

Agenda 8-4

Oracle Solaris 11 Network Implementation 8-5

Virtual Network Architecture 8-7

Virtual Network Topology 8-8

Agenda 8-9

Observing and Troubleshooting IP Interfaces and Datalinks 8-10

Verifying the Status of the Network Services 8-11

Verifying That the Network Daemon is Running 8-12

Verifying the Router and Protocols 8-13

Checking and Troubleshooting the DNS 8-15

Checking and Troubleshooting the DNS Server 8-16

Checking and Troubleshooting NFS 8-17

Checking Connectivity on an NFS Client 8-18

Checking the NFS Server Remotely 8-19

Verifying the NFS Service on the Server 8-21

Checking the Name-Service/Switch SMF Service 8-22

Quiz 8-23

Agenda 8-25

Using netstat to Display Network Information 8-26

Using netstat to Display Network Information: Example 8-28

Using ipadm to Display IP Interface and Address Information 8-29

Displaying Network Information: Example 8-31

Using dladm to Display Datalink Information 8-32

Displaying Datalink Information: Example 8-34

Using dlstat to Observe Network Traffic on Links 8-35

Using dlstat to Observe Network Traffic on Links: Example 8-36

Using flowstat to Observe Network Traffic on Flows 8-37

Using flowadm to Display Flow Information 8-38

Configure and Display Flow Information: Example 8-39

Using flowstat to Observe Network Traffic on Flows: Example 8-40

Quiz 8-41

Agenda 8-43

Case Study 7: Can't Access Network! 8-44

Case Study 7: Troubleshooting Steps 8-45
 Case Study 7: Walkthrough 8-46
 Additional Resources 8-54
 Practice 8: Overview 8-55
 Summary 8-56

9 Troubleshooting Workshop

Course Map 9-2
 Objectives 9-3
 Practice 9-4
 Summary 9-5

A Oracle Integrated Lights-Out Manager (ILOM)

Agenda A-2
 Introducing Oracle ILOM A-3
 Oracle ILOM: Overview A-4
 Identifying Key Oracle ILOM Features and Their Functionality A-5
 Identifying the Supported Management Interfaces A-8
 Getting Started A-10
 Agenda A-12
 Configuring Host Server Management Actions A-13
 Agenda A-15
 Collecting System Information and Monitoring Health Status A-16
 Collecting Information and Status A-17
 Viewing Open Problems A-19
 Agenda A-21
 Setting Up Alert Notifications and Syslog Server for Event Logging A-22
 Configuring Alert Notifications A-23
 Configuring Syslog for Event Logging A-24
 Agenda A-25
 Managing Event and Audit Log Entries A-26
 Agenda A-28
 Observing and Debugging System Behavior A-29
 Taking a Snapshot of the Oracle ILOM SP State (CLI) A-30
 Enabling SPARC Diagnostics to Run at Boot (CLI) A-32
 Agenda A-35
 Observing and Debugging System Behavior A-36
 Protecting Against Hardware Faults: Oracle ILOM Fault Manager A-37
 Using the Oracle ILOM Fault Management Shell A-39
 Launching a Fault Management Shell Session (CLI) A-40

Using fmadm to Administer Active Sun Hardware Faults	A-41
Clearing Faults for Undetected Replaced or Repaired Hardware Components	A-42
Using fmdump to View Historical Fault Management Logs	A-44
Using fmstat to View the Fault Management Statistics Report	A-46

Preface

Profile

Before You Begin This Course

Before you begin this course, you should be able to perform advanced Oracle Solaris 11 system administration tasks.

How This Course Is Organized

Oracle Solaris 11 Fault Analysis and Troubleshooting is an instructor-led course featuring lectures and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills that are introduced.

Related Publications

Oracle Publications

Title

N/A

Part Number

N/A

Additional Publications

- System release bulletins
- Installation and user's guides
- *read.me* files
- International Oracle User's Group (IOUG) articles
- *Oracle Magazine*

Typographic Conventions

The following two lists explain Oracle University typographical conventions for words that appear within regular text or within code samples.

1. Typographic Conventions for Words Within Regular Text

Convention	Object or Term	Example
Courier New	User input; commands; column, table, and schema names; functions; PL/SQL objects; paths	Use the <code>SELECT</code> command to view information stored in the <code>LAST_NAME</code> column of the <code>EMPLOYEES</code> table. Enter <code>300</code> . Log in as <code>scott</code>
Initial cap	Triggers; user interface object names, such as button names	Assign a When-Validate-Item trigger to the ORD block. Click the Cancel button.
Italic	Titles of courses and manuals; emphasized words or phrases; placeholders or variables	For more information on the subject see <i>Oracle SQL Reference Manual</i> Do <i>not</i> save changes to the database. Enter <i>hostname</i> , where <i>hostname</i> is the host on which the password is to be changed.
Quotation marks	Lesson or module titles referenced within a course	This subject is covered in Lesson 3, “Working with Objects.”

2. Typographic Conventions for Words Within Code Samples

Convention	Object or Term	Example
Uppercase	Commands, functions	SELECT employee_id FROM employees;
Lowercase, italic	Syntax variables	CREATE ROLE <i>role</i> ;
Initial cap	Forms triggers	Form module: ORD Trigger level: S_ITEM.QUANTITY item Trigger name: When-Validate-Item
Lowercase	Column names, table names, filenames, PL/SQL objects OG_ACTIVATE_LAYER (OG_GET_LAYER ('prod_pie_layer')) SELECT last_name FROM employees;
Bold	Text that must be entered by a user	CREATE USER scott IDENTIFIED BY tiger ;

1

Introduction

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Overview

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Welcome to the *Oracle Solaris 11 Fault Analysis and Troubleshooting* course. This specialized course is designed to teach fault analysis and troubleshooting skills to intermediate-level system administrators who are proficient in Oracle Solaris 11 system administration.

Course Goals

The goals of this course are to:

- Present you with a variety of troubleshooting techniques and then reinforce the use of those techniques through a series of examples that present potential issues with key features of the Oracle Solaris operating system
- Introduce you to both existing and new tools in Oracle Solaris 11 and show you how to use the tools through practical application
- Build your fault analysis and troubleshooting skills with numerous and meaningful practice opportunities

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Agenda: Day 1

- Lesson 1: Introduction
- Lesson 2: Analyzing System Failures and Troubleshooting Techniques
 - Identifying the basic layers of an Oracle Solaris system
 - Identifying the Oracle Solaris system error and fault types
 - Describing troubleshooting techniques
 - Gathering information related to a system fault
 - Mitigating the impact of system faults
 - Analyzing a case study that involves troubleshooting a system fault
- Lesson 3: Troubleshooting System Hardware Faults
 - Listing components commonly found in Oracle enterprise-class systems

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 11 Fault Analysis and Troubleshooting consists of five days of lecture and practices, with an emphasis on practice.

On the first day, Lesson 2 introduces you to the types of system faults you might encounter in Oracle Solaris 11 and a variety of troubleshooting techniques you can use to analyze the problem.

Lesson 3 focuses on the types of tools available to you to troubleshoot Oracle Solaris 11 system faults, such as the Service Management Facility (SMF) program, observability and administration tools, and debugger tools.

Course Agenda: Day 1

- Identifying faulty hardware behavior
- Identifying the best method for troubleshooting hardware faults
- Determining best tools for analyzing and troubleshooting hardware faults
- Describing how to use the Server Processor (SP) when troubleshooting hardware faults

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Agenda: Day 2

- Lesson 4: Troubleshooting Solaris Startup Faults
 - Identifying the stages associated with a successful Solaris OS startup
 - Listing faults common to Solaris OS startup failures
 - Identifying the best method for troubleshooting Solaris OS startup failure
 - Determining the best tools for analyzing and troubleshooting Solaris OS startup failures
 - Identifying the best tools and methods for analyzing and troubleshooting failures associated with system services
 - Determining when and how to use Solaris Boot Environments (BEs) for quick operating system recovery

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Lesson 4, you learn how to troubleshoot SPARC-based systems, with an emphasis on resolving booting problems.

Course Agenda: Day 2

- Lesson 5: Troubleshooting Image Packaging System (IPS) Problems
 - Checking the basics
 - Installing and updating packages
 - Fixing package problems
- Lesson 6: Troubleshooting Automated Installer (AI) Problems
 - Recalling the Automated Installer (AI) features and functions
 - Troubleshooting a client installation failure

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Lesson 5, you are shown how to troubleshoot IPS and packaging-specific problems.

In Lesson 6, you learn how to troubleshoot AI problems having to do with client installation failures and having to manually install after booting.

Course Agenda: Day 3

- Lesson 7: Troubleshooting Problems with Zones
 - Using zones-related SMF services
 - Monitoring non-global zones
 - Fixing IPS problems with non-global zones
- Lesson 8: Troubleshooting Physical and Virtual Network Problems
 - Checking the basics
 - Observing the network

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Lesson 7 covers how to troubleshoot problems with zones. In addition, you look at how to fix IPS problems with non-global zones.

In Lesson 8, you learn how to troubleshoot physical and virtual network problems by verifying the network configuration and observing the network.

Course Agenda: Days 4 and 5

- Lesson 9: Troubleshooting Workshop
 - Troubleshoot SPARC-based system boot problems.
 - Troubleshoot system hang problems.
 - Troubleshoot Automated Install problems.
 - Troubleshoot Image Packaging System problems.
 - Troubleshoot SMF services problems.
 - Troubleshoot network problems.
 - Analyze a system crash dump.
 - Troubleshoot zones problems.

Note: Class starts at 9 AM and ends at 5 PM each day. There are several short breaks throughout the day, with an hour for lunch.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Lesson 9, you put to practical use all the knowledge you gained in this course about troubleshooting techniques, tools, and technology-based information and use it to fix problems in every area of the system. During this two-day workshop part of the course, you are presented with a series of system faults and asked to diagnose them. You are given very little guidance as to where to look on the system, so these exercises are a true test of your troubleshooting skills.

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

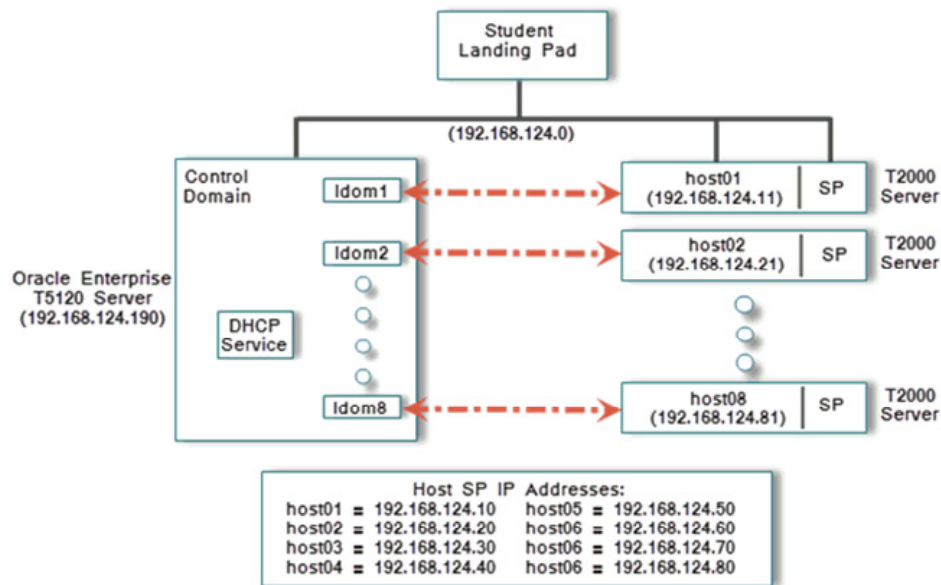
Your Learning Center

- 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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Your Practice Environment



ORACLE

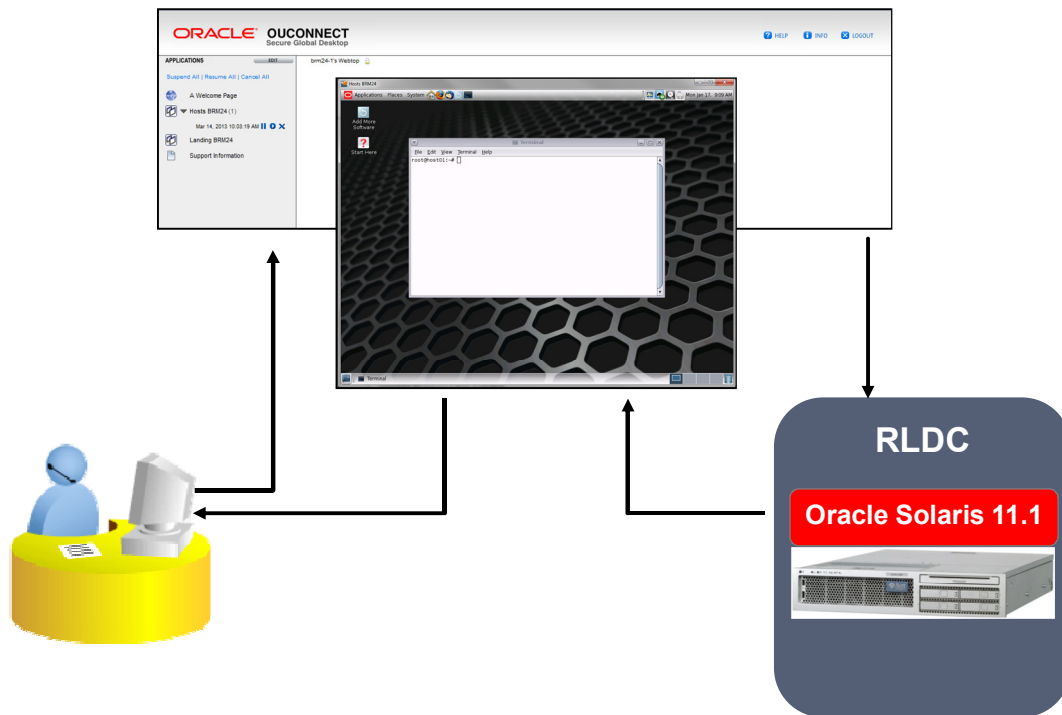
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The practice environment that you use in this course is based on SPARC T2000 server hardware running Oracle Solaris 11.1. and Oracle VM for SPARC logical domains (LDom). Each student will be assigned a T2000/LDom pair to use during this course. T2000/LDom pairing is shown in this illustration. For example: host01 is paired with Idom1, host02 is paired with Idom2, and so on.

Note: Each T2000 server is configured with 8 CPU cores, 16 GB of memory, two 72 GB disk drives, one physical NIC, and one console connection. Each server is configured with zones and VNICS.

Note: In the practice environment, each T2000 host contains a service processor (SP). When you require access to the system console, you telnet into your assigned host's SP. Additional instructions will be provided in your lab activity guides.

Accessing Your Practice Environment



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The practice environment that you use in this course is based on SPARC T2000 server hardware running Oracle Solaris 11.1. Because the hardware is physically located in Oracle's Remote Lab Data Center (RLDC), you access the practice environment by using Oracle Secure Global Desktop (SGD).

Each T2000 server is configured with 8 CPU cores, 16 GB of memory, two 72 GB disk drives, one physical NIC, and one console connection. Each server is configured with zones and VNICS.

Note: The practice environment also contains a number of service processors. When you require access to the console, you will telnet into one of these processors. You are given instructions on how to do this in the Oracle SGD environment later in this course.

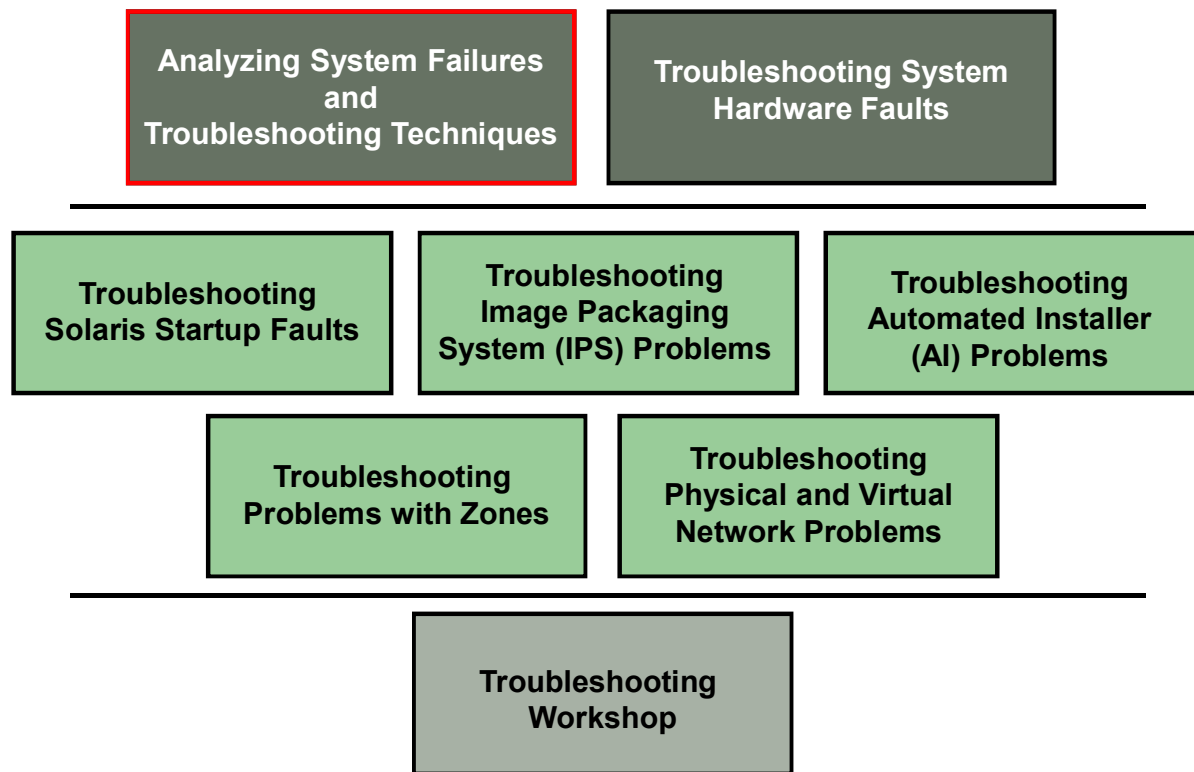
2

Analyzing System Failures and Troubleshooting Techniques

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The course map presented in the slide shows you the progress you are making in the course. The course map does not include Lesson 1 “Introduction”. The red highlight indicates the lesson you are currently looking at.

Objectives

After completing this lesson, you should be able to:

- Identify the basic layers of an Oracle Solaris system
- Identify Oracle Solaris system error and fault types
- Describe troubleshooting techniques
- Gather information related to a system fault
- Mitigate the impact of system faults
- Analyze a case study that involves troubleshooting a system fault

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

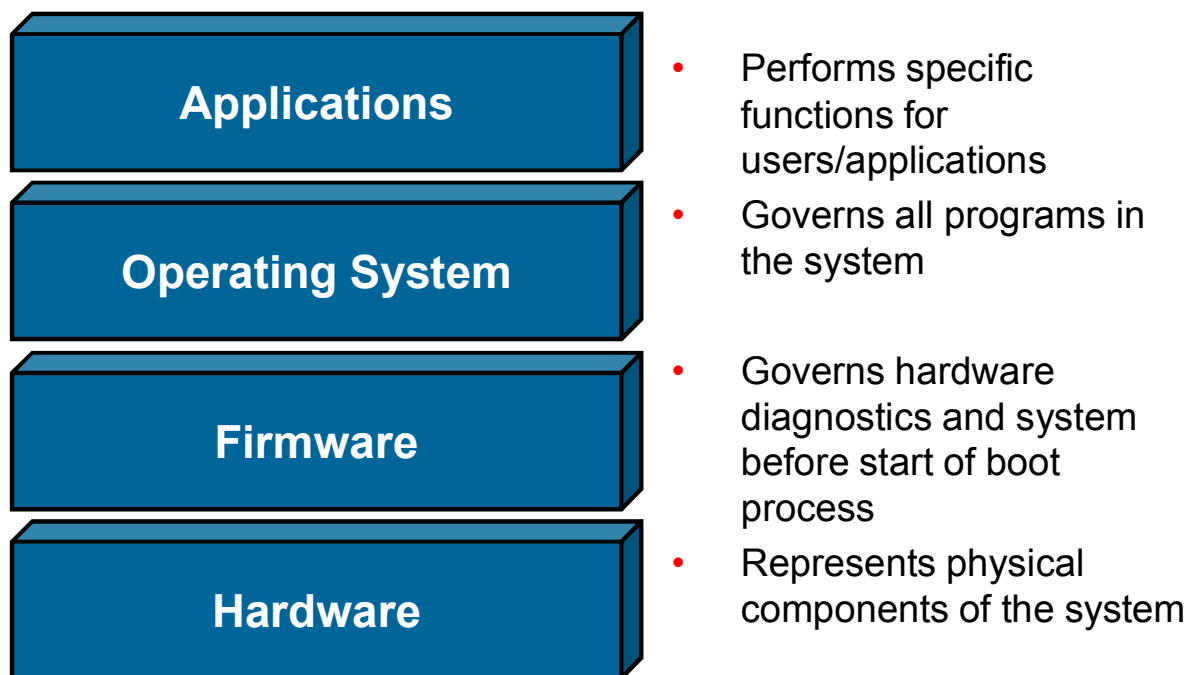
Agenda

- Oracle Solaris system fault types: Overview
- Troubleshooting strategy
- Gathering information related to the problem
- Mitigating fault impact
- Case study 1

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Identifying the Basic Layers of an Oracle Solaris System

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

An Oracle Solaris system has distinct system layers. Because a fault can occur in any system layer, to troubleshoot the system effectively, you must be familiar with all layers and how they interact with one another. An Oracle Solaris system has four basic layers (reading from bottom to top with regard to the graphic presented in the slide):

- **Hardware:** This layer represents the physical components of the system.
- **Firmware:** This layer governs hardware diagnostics and the system before the start of the boot process. Firmware is software that is embedded within special system chips. On many recently released platforms, a hypervisor module acts as the firmware layer between the hardware and the operating system. This module allows hardware updates without forcing compatibility updates to the operating system. The hypervisor accommodates changes to hardware and translates those changes to the operating system.
- **Operating System:** This layer governs all the programs in the system. The default operating system environment in Solaris systems is the Oracle Solaris operating system, which is designed for both SPARC and x86 architectures. This course focuses on the Oracle Solaris 11.1 operating system running on SPARC hardware.
- **Applications:** This layer performs a specific function for users or other applications, examples of which include databases and mail or web servers.

Identifying Error Types

Error Types	Short Description	Examples
Software	An error that does not originate in the hardware	<ul style="list-style-type: none"> • Programming errors in applications • Bugs in the kernel code
Hardware	An error that originates in the hardware	<ul style="list-style-type: none"> • Corrupt disks • Unexpected hardware interrupts
Critical	An error that results in a program or the system stopping	<ul style="list-style-type: none"> • Single power supply failure in a system with redundant power supplies • Fan failure that results in increased operating temperature

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

An Oracle Solaris system can have the following error types (as presented in the table in this slide and the next slide):

- **Software:** Software errors are errors that do not originate in the hardware. The system processor is involved in detecting certain software error types (for example, errors that cause an exception trap), but in some cases, the software detects errors on its own (for example, invalid state or checksum errors). Examples of software errors include programming errors in applications and bugs in the kernel code.
- **Hardware:** Errors that originate in the hardware. Examples include:
 - Corrupt disks
 - Power supply failures
 - Fan trays
 - Central processing unit (CPU) failures
 - Memory failures
 - Unexpected hardware interrupts

A hardware interrupt can indicate a hardware error. An interrupt is a signal that is generated by either a device that is attached to the system or a program within the system. The interrupt notifies the system of an event that can cause a program to suspend itself temporarily so that the CPU can process the relevant interrupt. Among hardware errors, an interrupt always signals hardware-corrected errors. No recovery action is usually required for this type of hardware error.

- **Critical error:** A critical error is an error that results in a program or the system stopping. Critical errors require immediate attention. It is recommended that the system be shut down immediately. Examples of critical errors include:
 - A single power supply failure in a system with redundant power supplies
 - Fan failure that results in an increased operating temperature

Identifying Error Types

Error Types	Short Description	Examples
Fatal	An error in which system recovery cannot be guaranteed	<ul style="list-style-type: none"> • Power supply problems in a system with a single power supply • Component burnout due to high temperatures
System Panic	An error that can corrupt data	<ul style="list-style-type: none"> • Hardware failure • Software bug in the operating system
Warning	An error that can impact some functionality	<ul style="list-style-type: none"> • Hardware failure such as a faulty fan • Software bug in a system service or application

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- **Fatal error:** A fatal error is an error in which system recovery cannot be guaranteed. Examples of fatal errors include:
 - Power supply problems in a system with a single power supply
 - Component burnout due to high temperatures
- **System panic:** A system panic occurs when the system detects a fatal error that can corrupt data. The system responds by halting all processes and calling the `panic()` kernel function. The `panic()` kernel function is not an error condition but a protective reaction to an error condition that is designed to safeguard system data. The `panic()` kernel function does the following:
 - Displays a panic message on the console
 - Performs a stack trace and lists routines that led to the panic
 - Generates a crash dump image of system memory in the dump device
 - Resets the system
- **Warning:** Warnings are generally caused by faults in non-critical hardware or software. This type of failure normally does not cause the system to panic but it could cause some loss of feature functionality.

Two-Phased Approach to Troubleshooting

- **Fault Analysis Phase**
 - State the problem.
 - Gather information.
 - Document the results of the fault analysis findings in a problem description.
- **Fault Diagnosis Phase**
 - Based on the fault analysis findings, determine the most probable causes of the fault.
 - Test and verify the probable causes.
 - Take corrective action.
 - Document the results of the fault diagnosis.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this course, you use a two-phased approach to troubleshooting. You begin with the fault analysis phase. In this phase, you state the problem, gather as much information about the problem as possible through such things as error messages, log files, documentation, user forums, and Oracle support websites, and document your findings in a problem description. Your problem description should identify what is and is not working.

In the fault diagnosis phase, you take the findings from the analysis phase to determine the most probable causes of the fault. You then test and verify your list of most likely causes. Through a process of elimination, you identify the actual cause of the fault while simultaneously verifying that you can correct the problem and not introduce any new problems. In the next slide, you are shown a number of approaches or techniques you can use to isolate the problem. After you have completed the test and verification step and have a fairly good idea of what caused the fault and how to safely correct it, you can take steps to actually correct the issue. The final step of the fault diagnosis phase is to document the steps you took to isolate and correct the problem.

Agenda

- Oracle Solaris system fault types: Overview
- **Troubleshooting strategy**
- Gathering information related to the problem
- Mitigating fault impact
- Case Study 1

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

High-Level Troubleshooting Strategy

- Clarify problem statement.
- Using the “assumptive” (best guess) method, prioritize the most probable causes of failure.
- Starting with the most probable cause of failure, gather pertinent information.
- Based on your analysis, quantify a solution for the fault.
 - Be sure to factor in the constraints imposed by the environment such as production machines.
- Apply the solution.
- Verify that the solution solves the problem.
- If the solution is not effective, try the next most probable cause of failure.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows a high-level troubleshooting strategy geared for quick fault resolution in production environments.

With the assumptive approach to troubleshooting, you rely less on documented sources and more on your knowledge of recent changes to the system and your own experience with similar problems.

Recent system changes are an important source of information about a system fault. For example, assume you are currently troubleshooting the problems of users who cannot log in to the system. You are aware that another system administrator was doing some maintenance work on the user files the day before. Based on past experience with diagnosing similar login issues, you make an educated guess that the system administrator accidentally removed read permissions from the `/etc` directory for files, such as `passwd` and `group`.

Your assumption is validated when you examine the files and find that the read permissions have indeed been removed.

Agenda

- Oracle Solaris system fault types: Overview
- Troubleshooting strategy
- **Gathering information related to the problem**
- Mitigating fault impact
- Case Study 1

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Clarifying the Problem Statement

Things to think about when a fault or error happens:

- When did the fault or error occur?
- What were you or other users on the system doing when the fault or error occurred?
- What functions can you not do?
- What functions can you still do?
- Can you duplicate the fault or error on another system?
- Can you duplicate the fault or error as another user or root?

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When a fault or error occurs, the first things to think about are how to locate the problem. The more information you have, the better—such as not only what you cannot do because the error occurred, but also what you can still do, and whether you can duplicate the problem on another system or as a different user.

Interpreting System Messages

```
# tail -f /var/adm/messages
Apr 13 10:11:07 host01 SC Alert: [ID 602306 daemon.error]
SYS_FAN at FT2 has FAILED.
Apr 13 10:15:37 host01 SC Alert: [ID 628137 daemon.alert]
BATTERY at SC/BAT/V_BAT has exceeded low warning threshold.
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The next important step is to gather as much information in the form of log files or system messages.

Checking system messages is a logical early step when trying to determine the probable cause of a system fault. At a glance, a system message can provide you with the following information:

- Process name/PID number
- Message ID number
- Facility that generated the message (for example, the kernel, a system daemon, or the `syslogd` daemon)
- Level of severity of the message (for example, emergency, error, warning, notice, or information)
- The message

Note: To view messages that are being sent to the `/var/adm` directory in real time, you can use the `tail -f /var/adm/messages` command. The `tail -f` command holds the file open so that you can view the messages being written to the file by the `syslogd` daemon.

In the slide is an example of two system messages, both triggered by a daemon error condition source. The first message has an error priority, and it is telling you that the system fan at FT2 has failed. The second message has an alert priority that is letting you know that the battery at SC/BAT/V_BAT has exceeded the low warning threshold.

From the information provided in the first message, you can call your service provider and schedule a server repair action when the server is not in use.

Because the second message has an alert priority, it requires immediate correction. From the information provided in the second message, you can call your service provider and schedule an immediate server repair action.

Gathering and Interpreting Log Files

- Install logs: `/var/sadm/system/logs/install_log`
- SMF logs: `/var/svc/log/<servicename>.log`
- AI: `/system/volatile/install_log`
- IPS: Run the `pkg history` command.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Log files are another important source of information when attempting to determine the cause of a problem.

You can find installation logs at `/var/sadm/system/logs/install_log`.

If an installation to a client system failed, you can find the log at `/system/volatile/install_log`.

With the `pkg history` command, you can check a history log of all package operations performed on your system.

Using Debug and Verbose Mode

- Many applications have a `-d/-D` option for debugging or a `-v/-V` option for verbose.
- In these modes, many more details are given about the executing routines, which can be very helpful for troubleshooting.
- For example, the `pkg verify` command in verbose mode shows each package that it verifies and displays it to the screen:

```
pkg verify -v
```
- For a host that is unreachable, try:

```
ping -v <hostname>
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Applications, programs, or commands often have a debug or verbose option. These are normally turned off, or not specified as the default, because the amount of information displayed in the output is usually not desired when everything is running smoothly. However, when a problem occurs, the detailed information can be useful when troubleshooting where the problem may lie.

Log File: Example

Gathering Solaris 11 installation error information

```
...
15:54:46 Creating IPS image
15:54:46 Error occurred during execution of 'generated-
transfer-1341-1' checkpoint. 15:54:47 Failed Checkpoints:
15:54:47 generated-transfer-1341-1
15:54:47 Checkpoint execution error:
15:54:47 Framework error: code: 6 reason: Couldn't resolve host
'pkg..mydomain.com' 15:54:47 URL:
'http://pkg.mydomain.com/solaris/release/versions/0/'.
15:54:47 Automated Installation Failed.
...
# more /system/volatile/install_log
...
TransportFailures: Framework error: code: 6 reason: Couldn't
resolve host 'pkg.mydomain.com' URL:
'http://pkg.mydomain.com/solaris/versions/0/'
...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Solaris 11 installation requires a number of technologies such as AI, IPS, Dynamic Host Configuration Protocol (DHCP), and so on. If an error occurs during install, the AI log `/system/volatile/install_log` is a good place to start your error-related data gathering.

In the slide you see an installation failure caused by AI failing to access the IPS service.

System Panics and Crash Dumps

- A panic occurs when the system encounters a fatal error caused by faults in the:
 - Operating system
 - Device drivers
 - OS loadable modules
 - Hardware
- When a panic occurs:
 - A message describing the error is usually echoed to the system console.
 - The system writes the contents of the physical memory to a predetermined dump device.
 - The dump is stored in two files: `unix.n` and `vmcore.n`.
 - The system is then rebooted.
- The crash dump provides invaluable information used by administrators to aid in diagnosing the problem.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A system panic occurs when the system encounters a fatal error caused by faults in the operating system, its associated device drivers and loadable modules, or by faulty hardware. When a panic occurs on a Solaris 11 system, a message describing the error is usually echoed to the system console. The system then attempts to write out the contents of physical memory to a predetermined dump device, which is usually a dedicated disk partition or the system swap partition. After this is completed, the system is then rebooted. The crash dump provides invaluable information used by administrators to aid in diagnosing the problem. When the system begins rebooting, a startup script calls the `savecore` utility, if enabled. This command performs a few tasks on the memory dump. First it checks to make sure that the crash dump corresponds to the running operating system. If the dump passes this test, `savecore` then begins to copy the crash dump from the dedicated dump device to the `/var/crash/`uname -n`` directory, or some other predetermined device. The dump is written out to two files, `unix.n` and `vmcore.n`, where `n` is a sequential integer identifying this particular crash. Finally, `savecore` logs a reboot by using the `LOG_AUTH syslog` facility.

Displaying the Current Crash Dump Configuration

To display the current crash dump configuration, use `dumpadm`.

```
# dumpadm
  Dump content: kernel pages
  Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash
Savecore enabled: yes
Save compressed: on
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To view the current dump configuration, use the `dumpadm` command without arguments, as shown in the slide.

Modifying the Crash Dump Configuration

To modify the crash dump configuration, use

```
/usr/sbin/dumpadm [-nuy] [-c content] [-d  
dump-device] [-m mink | minm | min%]  
[-s savecore-dir] [-r root-dir] [-z on | off].
```

```
# dumpadm -y -d /dev/dsk/c0t1d0s1  
    Dump content: kernel  
    Dump device: /dev/dsk/c0t1d0s1 (dedicated)  
Savecore directory: /var/crash  
Savecore enabled: yes  
Save compressed: on
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you want to modify the configuration of the crash dump file, you use the `dumpadm` command.

You can use several options with this command, as shown in the slide. The description for each option is as follows:

- `-n`: Specifies that `savecore` should not be run when the system reboots. Although this is the default setting, this dump configuration is not recommended. If system crash information is written to the swap device and `savecore` is not enabled, the crash dump information is overwritten when the system begins to swap.
- `-u`: Forcibly updates the kernel dump configuration based on the contents of the `/etc/dumpadm.conf` file. Normally, this option is used only on reboot when starting `svc:/system/dumpadm:default`, when the `dumpadm` settings from the previous boot must be restored. Your dump configuration is saved in the configuration file for this purpose.
- `-y`: Modifies the dump configuration to automatically execute the `savecore` command on reboot

- `-c content`: Specifies the type of data to dump. Use `kernel` to dump all kernel memory, `all` to dump all memory, or `curproc` to dump kernel memory and the memory pages of the process whose thread was executing when the crash occurred. The default dump content is kernel memory.
- `-d dumpdevice`: Specifies the device that stores the dump data temporarily when the system crashes. The primary swap device is the default dump device.
- `-m mink | minm | min%`: Specifies the minimum free disk space for saving crash dump files by creating a `minfree` file in the current `savecore` directory. This parameter can be specified in KB (`nnnk`), MB (`nnnm`), or file system size percentage (`nnn%`).
- `-s savecore-dir`: Specifies an alternative directory for storing crash dump files. The default `savecore-dir` directory is `/var/crash/hostname`, where the host name is the output of the `uname -n` command.
- `-r root-dir`: Specifies an alternative root directory relative to which the `dumpadm` command should create files. If the `-r` argument is not specified, the default root directory `/` is used.
- `-z on | off`: Modifies the dump configuration to control the operation of the `savecore` command on reboot. The `on` setting enables the saving of a core file in a compressed format. The `off` setting automatically uncompresses the crash dump file.

In the example in the slide, the kernel pages are dumped to a different dump device, `/dev/dsk/c0t1d0s1`, which is labeled as a dedicated dump device. In addition, the dump configuration is set to automatically execute the `savecore` command on reboot by using the `-y` option.

Saving the Crash Dump File

To save the crash dump file to the designated dump device, use `savecore -L`.

```
# savecore -L
dumping to /dev/dsk/c0t1d0s1, offset 65536, content:
kernel
  0:04 100% done
100% done: 103879 pages dumped, dump succeeded
savecore: System dump time: Tue Apr 16 10:23:31 2013

savecore: Saving compressed system crash dump in
/var/crash/client1/vmdump.0
savecore: Decompress the crash dump with
'savecore -vf /var/crash/vmdump.0'
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To save the contents of the crash dump file to the dump device that you have designated, use the `savecore -L` command. The `-L` option saves a crash dump of the live running Oracle Solaris system without actually rebooting or altering the system in any way. This option forces `savecore` to save a live snapshot of the system to the dump device, and then immediately to retrieve the data and to write it to a new set of crash dump files in the specified directory. Live system crash dumps can be performed only if you have configured your system to have a dedicated dump device by using the `dumpadm` command.

The `vmdump.0` file that you see in the example in the slide contains the recently created dump in compressed format.

Uncompressing the Crash Dump File

To uncompress the crash dump file, use `savecore -vf \var/crash/hostname/vmdump.0`.

```
# savecore -vf vmdump.0
savecore: System dump time: Tue Apr 16 10:25:20 2013
savecore: saving system crash dump in
/var/crash/{unix,vmcore}.0
Constructing namelist /var/crash/unix.0
Constructing corefile /var/crash/vmcore.0
 0:24 100% done: 103879 of 103879 pages saved
2266 (2%) zero pages were not written
0:24 dump decompress is done
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After you have saved the contents of the crash dump file to the dump device, you can uncompress the `vmdump.0` file by using the `savecore -vf` command, as shown in the slide. In the example in the slide, notice that this command (specifically the `-f` option) uncompresses the file to `vmcore.0`.

Analyzing the Crash Dump File Contents

To analyze the contents of the crash dump file, perform the following steps:

1. Change the directory to the `/var/crash` directory.
2. List the files in the crash directory.
3. Use the `mdb` command to analyze the crash dump file.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To analyze the contents of the crash dump files, you first need to go to the `/var/crash` directory. Next, you list the files that are in the directory. You should see these files listed: `bounds`, `unix.0`, `vmcore.0`, and `vmdump.0`. Use the `mdb` utility to analyze the crash dump file. You can use commands such as `strings`, `status`, or `cpuinfo` within `mdb`.

From this point, you send the crash dump files to an Oracle Solaris support engineer for analysis to determine what caused the system to crash.

mdb Essentials

- Launch the `mdb` utility: `mdb crash_dump_#`
- Get `mdb` help: `::help`
- List the currently loaded debugger modules: `::dmods`
- List the available debugger commands (`dcmds`): `::dcmds`
- List the walkers: `::walkers`

MDB Command Syntax:

```
[ address ] [ ,count ] command
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `mdb` utility is an extensible modular debugging utility. This utility provides an application programming interface (API) to compile modules and perform tasks within the context of the debugger. Before you use the `mdb` utility, you should be familiar with these terms:

- **Target:** The object to be inspected, such as a kernel crash dump or process core file, by the debugger
- **Debugger modules (dmods, pronounced dee-mods):** Dynamically loaded library that contains a set of debugger commands (`dcmds`) and walkers. During initialization, the `mdb` utility attempts to load the `dmods` that correspond to the load objects in the target. You can subsequently load or unload `dmods` at any time while running the `mdb` utility.
- **Debugger commands (dcmds, pronounced dee-commands):** A routine in the debugger that can access any properties of the current target. The `mdb` utility parses commands from the standard input and executes the corresponding `dcmds`. Each `dcmd` can also accept a list of string or numerical arguments.
- **Walker:** A set of routines that describe how to iterate through the elements of a particular program data structure. A walker encapsulates the implementation of a data structure from `dcmds` and the `mdb` utility. Use walkers interactively or use them to build other `dcmds` or walkers.

- **Walker:** A set of routines that describe how to iterate through the elements of a particular program data structure. A walker encapsulates the implementation of a data structure from `dcmds` and the `mdb` utility. Use walkers interactively or use them to build other `dcmds` or walkers.
- **Macros:** A text file that contains a set of commands that are used to automate the process of displaying commonly referenced programming structures:
 - `proc` macro: Displays the process structure
 - `thread` macro: Displays the thread structure
 - `inode` macro: Displays the inode structure

Note for the first bullet: When you launch the `mdb` utility, a response is displayed indicating which `mdb` modules are loaded.

Note for the second bullet: The `::help` command presents the general syntax for communicating with the debugger.

Note for the third bullet: To list the `dcmds` and walkers associated with a given debugger module, use `::dmods -l`.

Note about the MDG syntax: The *address* parameter is a kernel symbol. If the *address* parameter is not specified, the `mdb` utility uses the current location. The dot (`.`) also refers to the current location. If the *count* parameter is not specified, the `mdb` utility uses the default value of 1. The `mdb` utility commands consist of a verb followed by a modifier or a list of modifiers.

The following are the verbs that can be specified in the `mdb` command:

- `?:` Displays code or variables in an executable object file
- `/:` Displays data from the core file
- `==:` Prints values in different formats
- `$<:` Includes macro invocations for miscellaneous commands
- `>:` Assigns a value to a variable or a register
- `<:` Reads a value from a variable or a register
- **Return:** Repeats the previous command with a count of 1 and increments the current location represented by a dot (`.`)

Forcing a Crash Dump from OpenBoot PROM

1. Bring the system to the `ok` PROM prompt and type `sync`:

```
# init 0  
ok sync
```

2. Use the `halt -d` command:

```
# halt -d
```

3. Use the `reboot -d` command:

```
# reboot -d
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There might be times when you need to force a crash dump. First bring the system to the `ok` prompt. You do this by bringing the system to run-level 0 by typing `init0`. Then type `sync` at the `ok` prompt. Alternatively, you can use the `halt -d` or the `reboot -d` command to force a crash dump.

Forcing a Crash Dump on x86 (or SPARC): Example

```
# halt -d
Mar 28 11:30:12 host1 halt: halted by user
panic[cpu0]/thread=ffffffff83246ec0: forced crash dump
    initiated at user request
fffffe80006bbd60 genunix:kadmin+26b ()
fffffe80006bbec0 genunix:uadmin+163 ()
fffffe80006bbf10 unix:brand_sys_syscall+1dc ()

syncing file systems... done
dumping to /dev/zvol/dsk/rpool/dump offset 65536,
    content: kernel
0:06 100% done
100% done: 138934 pages dumped, dump succeeded
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to its upper right. The logo is white and is positioned on a solid red horizontal bar that spans the width of the slide.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows you how to force a crash dump on any type of platform (x86 or SPARC) by using the `halt -d` method.

Using kmdb to Force a Crash Dump

To boot a system with the kernel debugger (kmdb) enabled:

- Halt the system with `halt` command
- Type `boot -k` at the `ok` prompt:

```
ok boot -k
Boot device: /pci@7c0/pci@0/... File and args: -k
Loading kmdb. . .
```

- For run time, use the `mdb -k` command and to force a crash dump, use the `systemdump` macro:

```
[0] > $<systemdump
# savecore -vf /var/crash/vmdump.0
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can also use the kernel debugger (kmdb) to force a crash dump. The kernel debugger must have been loaded, either at boot time or with the `mdb -k` command for the following procedure to work.

1. You must be in text mode to access the kernel debugger. So first exit any window system.
2. Access the kernel debugger. The method that is used to access the debugger is dependent upon the type of console that you are using to access the system.
 - If you are using a locally attached keyboard, press F1 + A.
 - If you are using a serial console, send a break by using the method appropriate to that type of serial console.

The kmdb prompt is displayed.

3. To force a crash, use the `systemdump` macro.

```
[0] > $<systemdump
```

Panic messages are displayed, the crash dump is saved, and the system reboots.

4. Verify that the system has rebooted by logging in at the console login prompt.

Process Core Dumps

- What is a process core dump?
- Cause of process core dumps
- Core dump files
 - Per-process core file
 - Global core file
 - Zone core file

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A process core dump is a file that records the contents of a process along with other useful information, such as the processor register's value. When an application process receives a specific signal and terminates, the system generates a core dump and stops the process. In most cases, the signal leading to the application crash is SIGSEGV or SIGBUS. SIGSEGV indicates that the application is accessing an invalid memory address. SIGBUS indicates that the application is accessing a memory address that does not conform to CPU memory alignment rules. There are other signals whose default disposition is to create a core dump, for example, SIGFPE, which indicates a floating point exception.

The Solaris OS attempts to create up to three core dump files for each abnormally terminated process. One of the core dump files, which is called the per-process core file, is located in the current directory. Another core dump file, which is called the global core file, is created in the system-wide location. If the process is running in a local zone, a third core file is created in the global zone's location.

Managing Process Core Dumps

- `coreadm` command:

```
# coreadm
global core file pattern:
global core file content: default
init core file pattern: core
init core file content: default
global core dumps: disabled
per-process core dumps: enabled
global setid core dumps: disabled
per-process setid core dumps: disabled
global core dump logging: disabled
```

- Customize the process core dump configuration:

```
# coreadm -g /var/cores/%f.%n.%p.%t.core \
-e global \
-e global-setid \
-e log \
-d process \
-d proc-setid
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use the `coreadm` command to manage the core dumps. All the settings are saved in the `/etc/coreadm.conf` configuration file.

Running `coreadm` with no option shows you the current process's core dump configuration. The output shows:

- Global core dumps: A `disabled` line indicates no global core dump will be generated.
- Per-process core dumps: An `enabled` line indicates a per-process core dump will be generated for each abnormal process.
- `init core file pattern` line indicates the contents will be gathered from the live process to the per-process core dump.

You can use `coreadm` to customize the process core dump configuration, for example:

- The `-g` switch specifies the global core file name pattern. Unless a per-process pattern or setting overrides it, core files are stored in the specified directory with a name such as: `<program>.<node>.<pid>.<time>.core`

The `-e` switch specifies options to enable. The example shown is this slide enables the following options:

- Use of the global (system-wide) core file name pattern and location
- `setuid` programs also dump core as per the same pattern.
- Any attempt to dump core generates a syslog message.

The `-d` switch specifies options to disable. The example shown is this slide disables the following options:

- Core dumps per the per-process core file pattern.
- Per-process core dumps of `setuid` programs.

Manually Forcing Process Core Dumps

- **gcore utility:**

```
# gcore -o /var/tmp/myapp 392
```

- **gcore options and operand:**

- *-c content*
- *-F*
- *-g*
- *-o filename*
- *-p*
- *process-id*

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `gcore` utility creates a core image of each specified process. By default, the name of the core image file for the process whose process ID is the operand `process-id` is `core.process-id`. The following options are supported:

- *-c content*: Produces core image files with the specified content. The content description uses the same tokens as in `coreadm(1M)`. The *-c* option does not apply to cores produced due to the *-p* or *-g* flags.
- *-F*: Force. Grabs the target process even if another process has control.
- *-g*: Produces core image files in the global core file repository with the global content as configured by `coreadm(1M)`. The command fails if the user does not have permissions for the global core file repository.
- *-o filename*: Substitutes a file name in place of `core` as the first part of the name of the core image files. *filename* can contain the same tokens to be expanded as the paths in `coreadm`.
- *-p*: Produces a core image file in the process-specific location with the process-specific content for each process as configured by `coreadm`.

The operand `process-id` specifies the identifier of the process in the core dump.

Using gcore with pstack

1. Get the process ID of the suspect process:

```
# ps -eaf | grep -i suspect_process
```
2. Use the process ID to generate the gcore:

```
# gcore <proc_id>
```

A file `core.<proc_id>` is generated in the current directory.
3. Now generate the pstack based on the generated gcore file:

```
# pstack core.<proc_id>
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `pstack` command will print a hexadecimal dump of the process stack, thus providing important information about a suspect process on the affected system. Both the `gcore` and `pstack` commands are typically used together in a troubleshooting procedure.

Researching Previously Identified Problems

- Solaris 11 Release Notes
- My Oracle Support (MOS):
 - Search for solutions.
 - Access proactive support tools.
 - Collaborate in the My Oracle Support Community.
 - Create a service request.
- Oracle Technology Network (OTN):
 - Download software.
 - Access documentation, technical articles, blogs, discussion forums, known patch and bug descriptions, wikis, and sample code.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Solaris 11 Release Notes is a valuable resource for diagnosing existing problems identified by the Solaris engineering teams at product release time. Also, the Internet and Oracle websites, such as My Oracle Support (MOS) and the Oracle Technology Network (OTN) are other options for researching problems and finding possible solutions. MOS is specifically designed to provide support for Oracle Premier Support Customers whereas OTN is free to join and is intended for the software developer, system administration, and solution architect communities.

Agenda

- Oracle Solaris system fault types: Overview
- Troubleshooting strategy
- Gathering information related to the problem
- **Mitigating fault impact**
- Case study 1

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Mitigating Fault Impact

- Using Oracle Configuration Manager (OCM) and Auto Service Request (ASR)
- Running Oracle Explorer or other data-collecting scripts
- Using the Fault Management Architecture (FMA) resilient error handling
- Using alternate Boot Environments (BEs) and ZFS snapshots as a safeguard against downtime associated with software updates and system reconfiguration

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

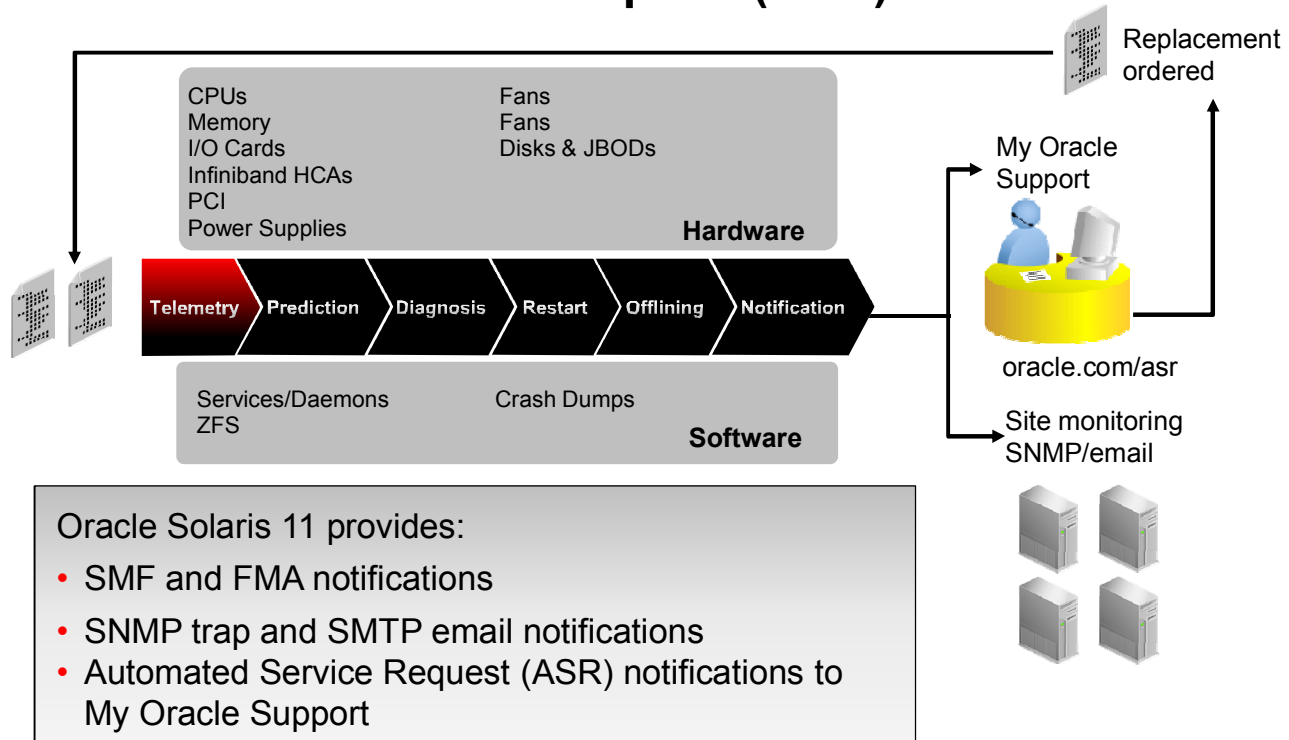
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 11 provides you with a variety of important features that are designed to reduce the time required for identifying and repairing system faults, and storing system functionality. These include: automated support services using Oracle Configuration Manager (OCM) and Auto Service Request (ASR), downloadable system assessment tools such as Oracle Explorer, resilient error handling using Fault Management Facility, and minimizing downtime associated with software updates and system reconfiguration.

Note: If the SMF properties needed for registration are not available to the ASR SMF service (`asr-notify`), a message is sent to the system log that asks the user to register the system. The `asr-notify` service sleeps and waits for the registration properties to be provided by the `asradm` command.

When you register for OCM and ASR support, your system's configuration data is collected and uploaded to Oracle's customer configuration repository. The data is analyzed by Oracle's customer support representatives to provide proactive problem detection and faster problem resolution for support issues. The configuration information is also used to support upgrades, enable accurate planning, and track configuration changes.

Using SMF, FMA, SNMP, SMTP, and Auto Service Request (ASR) Notifications



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 11.1 has the ability to notify administrators of SMF service state changes and FMA fault management events. Administrators can configure Simple Network Management Protocol (SNMP) trap notifications and Simple Mail Transfer Protocol (SMTP) email notifications to watch certain events or services. Notifications can also be sent through to Oracle with Automated Service Request (ASR) notifications, providing automatic telemetry to customers who have an active Oracle Support agreement. When an ASR notification is received, Oracle Support can automatically begin ordering a replacement part and ensuring that the replacement part is delivered and installed during a planned downtime opportunity. For more information about the ASR process, see the oracle.com/asr website.

Running Oracle Explorer or Other Data-Collecting Scripts

Oracle Services Tools Bundle for Sun Systems:

- **Oracle Explorer Data Collector:** Provides a detailed snapshot of system configurations and states for diagnosis and troubleshooting
- **Oracle Lightweight Availability Collection Tool:** Tracks system availability data such as boot, panic, and halt events
- **Oracle Remote Diagnostic Agent:** Collects essential system information for Oracle Support
- For more information, see:
 - <http://www.oracle.com/us/support/systems/premier/services-tools-bundle-sun-systems-163717.html>
 - Knowledge Doc ID 1153444.1: <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1153444.1>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Customers who have Oracle Premier Support for Systems or a valid Sun support plan can download the Oracle Services Tools Bundle for Sun Systems, which is a self-extracting installer bundle that supports the Oracle Solaris operating system and architecture. The bundle contains a number of tools that enable both Oracle service engineers and customers to diagnosis and solve problems and proactively prevent future issues. The tools include but are not limited to:

- **Oracle Explorer* Data Collector:** A collection of shell scripts and binary executables that gathers information related to drivers, patches, recent system event history, and log file entries and creates a detailed snapshot of the system configurations and states. The Explorer output enables Oracle engineers to assess Sun systems by applying the output against a knowledge-based rules engine. There is a comprehensive user's guide available for customers who want to use the tool for their own problem diagnosis. For access to the guide, see the Knowledge Doc link provided in the slide.
*Explorer should now be available via IPS on Solaris 11.1.
- **Oracle Lightweight Availability Collection Tool:** Designed to simplify Oracle's availability collection process for Sun systems and accommodate customer network and security policies, this lightweight stand-alone tool allows the user to track system availability data, such as boot, panic, and halt events.

- **Oracle Remote Diagnostic Agent:** A unified package of support diagnostics tools and preventive solutions that collects essential system information. The data that is captured provides Oracle Support with a comprehensive picture of the customer's environment, which aids in problem diagnosis.

For more information about the Oracle Services Tools Bundle, including component software downloads and frequently asked questions, log in to My Oracle Support and reference Knowledge Doc ID 1153444.1 (a link to which is provided in the slide). The knowledge document also contains instructions for installing the bundle and specific information about how to use the Oracle Explorer Data Collector.

Note: If you are an Oracle support customer and have not registered on My Oracle Support, visit the My Oracle Support Welcome Center for complete registration instructions.

FMA Features

The Fault Management stack contains:

- Error detectors
- Diagnosis engines
- Response agents



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle Solaris Fault Management feature provides an architecture for building resilient error handlers, structured error telemetry, automated diagnostic software, response agents, and structured messaging. Many parts of the software stack participate in Fault Management including the CPU, memory and I/O subsystems, Oracle Solaris ZFS, a growing set of device drivers, and other management stacks.

FMA is intended to help with problems that can occur on an Oracle Solaris system. The problem could be a fault, meaning something that used to work but no longer does. The problem could be a defect, meaning that it never worked correctly. In general, hardware can experience both faults and defects. However, most software problems are defects or are caused by configuration issues.

At a high level, the Fault Management stack contains error detectors, diagnosis engines, and response agents. Error detectors, as the name suggests, detect errors in the system and perform any immediate required handling. Error detectors issue well-defined error reports, or ereports, to a diagnosis engine. A diagnosis engine interprets ereports and determines whether a fault or defect is present in the system. When such a determination is made, the diagnosis engine issues a suspect list that describes the resource or set of resources that might be the cause of the problem. The resource might or might not have an associated field-replaceable unit (FRU), a label, or an Automatic System Reconfiguration Unit (ASRU). An ASRU may be immediately removed from service to mitigate the problem until the FRU is replaced.

When the suspect list includes multiple suspects (for example, if the diagnosis engine cannot isolate a single suspect), the suspects are assigned a probability as to each suspect being the key suspect. The probabilities in this list add up to 100 percent. Suspect lists are interpreted by response agents. A response agent attempts to take some action based on the suspect list. Responses include logging messages, taking CPU strands offline, retiring memory pages, and retiring I/O devices.

Error detectors, diagnosis engines, and response agents are connected by the Fault Manager daemon, `fmd`, which acts as a multiplexor between the various components, as shown in the figure in the slide.

The Fault Manager daemon is itself a service under SMF control. The service is enabled by default and controlled just like any other SMF service. The FMA and SMF services interact with each other when appropriate. Certain hardware problems can cause services to be stopped or restarted by SMF. Also, certain SMF errors cause FMA to report a defect.

Interpreting Notifications of Faults and Defects

1	# fmadm faulty			
2	-----	-----	-----	-----
3	TIME	EVENT-ID	MSG-ID	SEVERITY
4	-----	-----	-----	-----
5	Apr 16 17:56:03	7b83c87c-78f6-6a8e-fa2b-d0cf16834049	SUN4V-8001-8H	Minor
6				
7	Host	: bur419-61		
8	Platform	: SUNW,T5440	Chassis_id : BEL07524BN	
9	Product sn	: BEL07524BN		
10				
11	Fault class	: fault.cpu.ultraSPARC-T2plus.ireg		
12	Affects	: cpu:///cpuid=0/serial=1F95806CD1421929		
13		faulted and taken out of service		
14	FRU	: "MB/CPU0" (hc:///product-id=SUNW,T5440:server-id=bur419-61:\		
15		serial=3529:part=541255304/motherboard=0/cpuboard=0)		
16		faulty		
17	Serial ID.	: 3529		
18		1F95806CD1421929		
19				
20	Description	: The number of integer register errors associated with this thread		
21		has exceeded acceptable levels.		
22				
23	Response	: The fault manager will attempt to remove the affected thread from		
24		service.		
25				
26	Impact	: System performance may be affected.		
27				
28	Action	: Use 'fmadm faulty' to provide a more detailed view of this event.		
29		Please refer to the associated reference document at		
30		http://support.oracle.com/msg/SUN4V-8001-8H for the latest service		
31		procedures and policies regarding this diagnosis.		

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Often, the first interaction with the Fault Manager daemon (`fmd`) is a system message indicating that a fault or defect has been diagnosed. Messages are sent to both the console and the `/var/adm/messages` file. All messages from the Fault Manager daemon use the format presented in the slide.

When notified of a diagnosed problem, always consult the recommended knowledge article for additional details. See line 13 in the slide for an example. The knowledge article might contain additional actions that you or a service provider should take beyond those listed on line 11.

Notification of Fault Manager error events can be configured by using SNMP or SMTP. You are shown how to set up this notification in the lesson titled "Troubleshooting System Hardware Faults".

In addition, ASR can be configured to automatically request Oracle service when specific hardware problems occur. See the Oracle Auto Service Request product page for information about this feature.

Gathering and Interpreting FMA Data by Using `fmadm`, `fmstat`, and `fmdump`

`fmd` daemon information gathering utilities:

- `fmadm`: Used to view and modify system configuration parameters maintained by Solaris Fault Manager
- `fmstat`: Used to report statistics associated with the Solaris Fault Manager, `fmd(1M)`, and its associated set of modules
- `fmdump`: Used to display the contents of any of the log files associated with the Solaris Fault Manager

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The fault management daemon (`fmd`) is a daemon that runs in the background on each Solaris system. `fmd` receives telemetry information relating to problems detected by the system software, diagnoses these problems, and initiates proactive self-healing activities such as disabling faulty components. When appropriate, the Fault Manager also sends a message to the `syslogd(1M)` service to notify an administrator that a problem has been detected.

The `fmd` daemon provides three robust information gathering utilities:

- **`fmadm`**: Used to view and modify system configuration parameters maintained by Solaris Fault Manager, `fmd`. You can use this utility to:
 - View the set of diagnosis engines and agents that are currently participating in fault management.
 - View the list of system components that have been diagnosed as faulty.
 - Perform administrative tasks related to these entities.
- **`fmstat`**: Used to show statistics about each module active in FMA. When a module is specified to the `fmstat` command, all statistics for that module are shown.

- **fmdump:** Used to show FMA error and fault log information. These logs are in the `/var/fm/fmd` directory.

Note: The Fault Manager daemon `fmd` records information in several log files. The log files are stored in `/var/fm/fmd` and are viewed by using the `fmdump` command.

- The `errlog` log file records inbound telemetry information that consists of ereports.
- Informational events are recorded in two log files. `info_log_hival` is for high-value events, and `info_log` collects all other informational events.
- The `fltlog` log file records fault diagnosis and repair events.

Caution: Do not base administrative action on the contents of the log files, but rather on the `fmadm faulty` output. The log files can contain error statements that should not be considered faults or defects.

Gathering and Interpreting FMA Data by Using `fmadm`: Example

1	# <code>fmadm faulty</code>		
2	-----	-----	-----
3	TIME	EVENT-ID	MSG-ID SEVERITY
4	-----	-----	-----
5	Aug 24 17:56:03	7b83c87c-78f6-6a8e-fa2b-d0cf16834049	SUN4V-8001-8H Minor
6			
7	Host	: bur419-61	
8	Platform	: SUNW,T5440	Chassis_id : BEL07524BN
9	Product_sn	: BEL07524BN	
10			
11	Fault class	: fault.cpu.ultraSPARC-T2plus.ireg	
12	Affects	: cpu:///cpuid=0/serial=1F95806CD1421929	
13		faulted and taken out of service	
14	FRU	: "MB/CPU0" (hc:///product-id=SUNW,T5440:server-id=bur419-61:\	
15		serial=3529:part=541255304/motherboard=0/cpuboard=0)	
16		faulty	
17	Serial ID.	: 3529	
18		1F95806CD1421929	
19			
20	Description	: The number of integer register errors associated with this thread	
21		has exceeded acceptable levels.	
22			
23	Response	: The fault manager will attempt to remove the affected thread from	
24		service.	
25			
26	Impact	: System performance may be affected.	
27			
28	Action	: Use 'fmadm faulty' to provide a more detailed view of this event.	
29		Please refer to the associated reference document at	
30		http://support.oracle.com/msg/SUN4V-8001-8H for the latest service	
31		procedures and policies regarding this diagnosis.	

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The preferred method to display fault or defect information and determine the FRUs involved is the `fmadm faulty` command. However, the `fmdump` command is also supported. `fmdump` is often used to display a historical log of problems on the system, and `fmadm faulty` is used to display the active problems.

In the example in the slide, line 14 is of primary interest, which shows the data for the impacted FRUs. The more human-readable location string is presented in quotation marks, "MB/CPU0". The quoted value is intended to match the label on the physical hardware. The FRU is also represented in a Fault Management Resource Identifier (FMRI) format that includes descriptive properties about the system containing the fault, such as its host name and chassis serial number. On platforms that support it, the part number and serial number of the FRU are also included in the FRU's FMRI.

The `Affects` lines (lines 12 and 13) indicate the components that are affected by the fault and their relative state. In this example, a single CPU strand is affected. It is faulted and taken out of service. Following the FRU description in the `fmadm faulty` command output, line 16 shows the state as `faulty`. The `Action` section might also include other specific actions instead of, or in addition to, the usual reference to the `fmadm` command.

Gathering and Interpreting FMA Data by Using `fmstat`: Example

```
# fmstat
module          ev_recv ev_acpt wait  svc_t  %w  %b  open  solve  memsz  bufsz
cpumem-retire    1        0  0.0  403.5  0   0    0     0    419b    0
disk-transport   0        0  0.0  500.6  0   0    0     0    32b     0
eft              0        0  0.0   4.8   0   0    0     0    1.4M   43b
fmd-self-diagnosis 0        0  0.0   4.7   0   0    0     0     0     0
io-retire        0        0  0.0   4.5   0   0    0     0     0     0
snmp-trapgen     0        0  0.0   4.5   0   0    0     0    32b     0
sysevent-transport 0        0  0.0 1444.4 0   0    0     0     0     0
syslog-msgs      0        0  0.0   4.5   0   0    0     0     0     0
zfs-diagnosis    0        0  0.0   4.7   0   0    0     0     0     0
zfs-retire       0        0  0.0   4.5   0   0    0     0     0     0
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Without options, `fmstat` gives a high-level overview of the events, processing times, and memory usage of the loaded modules, an example of which is presented in the slide. The columns of data are:

- **ev_recv:** Number of events received by the module
- **ev_acpt:** Number of events accepted by the module as relevant to a diagnosis
- **wait:** Average number of events waiting to be examined by the module
- **svc_t:** Average service time in milliseconds for events received by the module
- **%w:** Percentage of time that there were events waiting to be examined by the module
- **%b:** Percentage of time that the module was busy processing events
- **open:** Number of active cases owned by the module
- **solve:** Total number of cases solved by this module since it was loaded
- **memsz:** Amount of dynamic memory currently allocated by this module
- **bufsz:** Amount of persistent buffer space currently allocated by this module

You can display statistics on an individual module by using the `-m` module option. This syntax is commonly used with the `-z` option to suppress zero-valued statistics.

Gathering and Interpreting FMA Data by Using `fmdump`: Example

```

1  % fmdump -v -u 7b83c87c-78f6-6a8e-fa2b-d0cf16834049
2  TIME                               UUID                               SUNW-MSG-ID EVENT
3  Apr 16 17:56:03.4596 7b83c87c-78f6-6a8e-fa2b-d0cf16834049 SUN4V-8001-8H Diagnosed
4    100%  fault.cpu.ultraSPARC-T2plus.ireg
5
6  Problem in: -
7    Affects: cpu:///cpuid=0/serial=1F95806CD1421929
8             FRU: hc//:product-id=SUNW,T5440:server-id=bur419-61:\
9             serial=9999:part=541255304/motherboard=0/cpuboard=0
10            Location: MB/CPU0

```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Some console messages and knowledge articles might instruct you to use the older `fmdump -v -u UUID` command to display fault information. Although the `fmadm faulty` command is preferred, the `fmdump` command still operates, as shown in the example in the slide. The information about the affected FRUs is still present, although separated across three lines (lines 8 through 10). The `Location` string presents the human-readable FRU string. The `FRU` lines present the formal FMRI. Note that the severity, descriptive text, and action are not shown with the `fmdump` command unless you use the `-m` option. See the `fmdump(1M)` man page for more information.

Using ZFS Snapshots or Alternate Boot Environments (BEs) as a Safeguard

This section covers the following topics:

- Using ZFS snapshots for ZFS data backup and recovery
- Using alternate boot environments (BEs) for backup and testing purposes

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

File System Backup and Recovery with Snapshots

- You can use ZFS snapshots to:
 - Take backups of your local file systems (`zfs snapshot` or `zfs snapshot -r`)
 - Recover a file system by rolling back a snapshot (`zfs rollback`)
- To view snapshots, you must enable the `listsnapshots` pool property (`zpool listsnapshots=on pool`).
- When you roll back a snapshot:
 - All changes made to a file system since a specific snapshot was created are discarded.
 - All intermediate snapshots and their clones must be destroyed (`zfs rollback -rR`).
 - The file system that you want to roll back must be unmounted.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you do not have a business requirement to store your data remotely for disaster recovery purposes, you can take snapshots of your local file systems as backups and then, if necessary, recover them by using the ZFS snapshot rollback feature.

You can use the `zfs snapshot` command to create a snapshot, which takes the name of the snapshot that you want to create as its only argument. The snapshot name must be in one of these two formats: *filesystem@snapname* or *volume@snapname*. You can also create a recursive snapshot by using the `-r` option. To view the snapshots that you created, you must enable the `zpool listsnapshots` property for the pool in which the file systems reside. This property is disabled by default.

The `zfs rollback` command discards all changes made to a file system since a specific snapshot was created. The file system reverts to its state at the time the snapshot was taken. By default, the command cannot roll back to a snapshot other than the most recent snapshot. For the rollback to occur, all intermediate snapshots must be destroyed. You can destroy earlier snapshots by specifying the `-r` option. If clones of any intermediate snapshots exist, you must destroy them as well by specifying the `-R` option.

Note: The `-rR` options do not recursively destroy the child snapshots of a recursive snapshot. Only the top-level recursive snapshot is destroyed by either of these options. To completely roll back a recursive snapshot, you must roll back the individual child snapshots.

Before you roll back a file system, you must ensure that it is unmounted. If not, the rollback fails. If necessary, you can force the file system to be unmounted by using the `zfs rollback -f` command. After the rollback has completed, you can remount the file system.

Using Snapshots for Backup and Recovery

1. Use `zpool get listsnapshots poolname` to determine a pool's snapshot property setting.
2. Use `zpool set listsnapshots=on poolname` to enable a pool's snapshot property.
3. Use `zpool get listsnapshots poolname` to verify that a pool's snapshot property is enabled.
4. Use `zfs snapshot filesystem@snapshot` to create a snapshot of a file system.
5. Use `zfs list -r filesystem` to view the snapshots.
6. Use `zfs rollback filesystem@snapshot` to recover a file system, if necessary.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use snapshots to back up your file systems and, if necessary, recover a file system using the `zfs rollback` command. The steps for enabling the `zpool listsnapshots` property and using snapshots for file system backup and recovery are presented in the slide.

Notes for step 4: You can also create a recursive snapshot of a file system by using the `zfs snapshot -r filesystem@snapshot` command.

Notes for step 6: If intermediate snapshots exist for the file system, you must destroy them along with any clones of the intermediate snapshots by using the `zfs rollback -rR filesystem@snapshot` command.

Using Alternate Boot Environments (BEs)

- Use an alternate boot environment as a backup and as a place to test and validate potential fixes.
- Use a snapshot of an existing boot environment for reference or to create a new boot environment.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Creating a backup of your existing boot environment gives you the flexibility of having an environment in which to test and validate changes as part of your troubleshooting activities instead of having to work in the production environment. After you have tested and validated your proposed fixes to a problem, you can either make the alternate boot environment the active boot environment or you can make the fixes directly in the original boot environment.

Another thing you can do to facilitate your troubleshooting activities is to manually create a snapshot of an existing boot environment for reference. This snapshot is a read-only image of a data set or boot environment at a given point in time.

You also have the option of creating a new boot environment from an existing snapshot. Then you can activate and boot that new boot environment.

Alternate Boot Environments: Example

To create an alternate boot environment, use the `beadm` command as seen below. Then you can activate it and it will be active on the next `reboot`:

```
# beadm create test_be
# beadm activate test_be
# beadm list
```

BE	Active	Mountpoint	Space	Policy	Created
--	----	-----	-----	-----	-----
test_be	R	-	9.89G	static	2000-02-24 13:06
solaris	N	/	51.64M	static	2000-02-08 15:17
solaris.orig	-	-	1.17M	static	2000-02-08 17:23

```
# reboot
# beadm list
```

BE	Active	Mountpoint	Space	Policy	Created
--	----	-----	-----	-----	-----
test_be	NR	/	9.89G	static	2000-02-24 13:06
solaris		-	51.64M	static	2000-02-08 15:17
solaris.orig	-	-	1.17M	static	2000-02-08 17:23

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can create multiple boot environments that may be helpful for setting up test environments or as a safeguard. You can also boot to an alternate boot environment at the PROM level by using the `boot -Z rpool/ROOT/test_be` command at the `ok` prompt.

Agenda

- Oracle Solaris system fault types: Overview
- Troubleshooting strategy
- Gathering information related to the problem
- Mitigating fault impact
- Case study 1

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 1: System Panic

- Clarify the problem statement.
 - System `host1` has experienced a system panic!
- Using “assumptive” method, prioritize the most probable causes of failure.
 - Recent software update
 - Faulty hardware
 - Configuration error
- Starting with the most probable cause of failure, gather pertinent information.
 - Peruse system message log.
 - Run the `mdb` utility on crash dump.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The next series of slides walk you through a case study involving a system panic. The strategy used in the study is based on the troubleshooting strategy discussed earlier in this lesson.

Case Study 1: Troubleshooting Steps

The following steps are used to identify the cause of the crash:

1. Search the `/var/adm/messages` file for error messages.
2. Invoke the `mdb` utility on the most recent `unix.n` and `vmcore.n` files by using `mdb x`, where `x` indicates the number of the latest crash dump.
3. Examine the general status information about the dump by using the `::status` command.
4. Use `::msgbuf` to display console output.
5. Dump the values of the registers at the time of the panic.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When a problem occurs, the first step is to check the messages file for error indications. You use the `mdb` utility to conduct a postmortem analysis of the Solaris OS kernel crash dumps.

As a system administrator without a programming background, a manageable set of objectives for crash analysis is to use the `mdb` utility to perform the following:

- Identify the address of the instruction that caused the panic.
- Identify the address of the thread that was running during the panic.
- Identify the name and arguments of the processes that were running during the panic.

You use the `mdb` command to view the contents of the processor registers to extract information that can assist in determining the cause of a system crash. Registers contain information about the status of the system.

Assume you have already done the following:

- Configured the system crash facilities
- Saved the crash dump file to the designated dump device by using `savecore -L`
- Uncompressed the crash dump file by using `savecore -vf /var/crash/hostname/vmdump.0`

You are now ready to use `mdb` to examine the cause of the panic. Follow the steps presented in the slide to complete this task.

Note for step 5: The output may scroll for several pages. In general, you are interested in the end of the output, that is, where the panic occurred.

Note for step 6: The usefulness of the register values is determined by the type of problem you are examining. In a system dump analysis, the most important registers are the program counter and the stack pointer. The program counter, `o7` or `pc`, contains the address of the current instruction, and the stack pointer, `o6` or `sp`, points to the current stack frame for use with local variables or return addresses. Use the debugger `$r` command to display the registers and some specific pointers.

Case Study 1: Troubleshooting Steps

6. Examine the instruction that caused the panic by using the following command: `<pc::dis`.
7. Examine the thread that was running at the time of the panic by using the `panic_thread` value.
8. Use the `t_procp` member to get to the command-line arguments in effect at the time of the panic.
9. Use `$c` to generate a stack backtrace to show how the system arrived at the bad location.
10. Use `::cpuinfor -v` to obtain information about what was running on the CPUs at the time of the panic.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note for step 7: The `::dis` command makes the instruction corresponding to the specified address (in this case, the value of the program counter register) bold. You use the program counter register and disassemble the contents at the address, plus 10 instructions before and after the address.

Note for step 8: The `panic_thread` value is a pointer to the thread that was running when the panic occurred. You could also use global register 7 (`<g7::print kthread_t`).

Note for step 9: Generally, the most useful information in the `kthread_t` is the pointer to the `proc_t` for the process in which the thread is running. Depending on the nature of the panic, any of the other fields might also be interesting. One reason that you might want to know the command-line arguments that were running at the time of the panic is to determine if the panic is reproducible.

For more information about `mdb`, see the *Oracle Solaris Modular Debugger Guide* and the `mdb` man page.

Case Study 1: Walkthrough

```
# cat /var/adm/messages | egrep '(panic)|(trap)'
panic[cpu0]/thread=30002c57b00:
BAD TRAP: type=10 rp=2a100bb3500 addr=1397814 mmu_fsr=0
# cd /var/crash/'uname -n'
# ls
bounds          unix.1          unix.3          vmcore.1        vmcore.3
unix.0          unix.2          vmcore.0        vmcore.2
# mdb 3
Loading modules: [ unix genunix specfs dtrace zfs scsi_vhci sd mpt mac px ldc ip
hook neti arp usba kssl fctl sockfs random mdesc idm cpc crypto fcip fcp ufs
logindmux nsmb ptm spps nfs lofs ipc ]
> ::status
debugging crash dump vmcore.3 (64-bit) from host1
operating system: 5.1 11.1 (sun4v)
usr/src version: 18481:2838b1a36220:on11u1_24b+1
usr/closed version: 1703:0d660c5b838a:on11u1_24b+1
image uuid: 21c1c4c0-c591-e89c-aad0-f4d95c1fc91d
panic message: BAD TRAP: type=10 rp=2a100bb3500 addr=1397814 mmu_fsr=0
dump content: kernel pages only
> ::msgbuf
MESSAGE
dad1 is /pci@1f,0/pci@1,1/ide@3/dad@0,0
root on rpool/ROOT/s11.1 fstype zfs
PCI-device: ebus@1, ebus0
Many lines of output omitted...
/pci@1f,0/pci@1/pci@2/usb@8,1/mouse@2 (hid0) online
>> More [<space>, <cr>, q, n, c, a] ? c ? continue
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in the slide, the cause of a crash dump is being examined by using the steps discussed in the previous slides. The first step is to change to the `savecore` directory. You then list the `savecore` directory contents to note the most recent crash dump files. As you can see, this is the fourth crash dump this system has recorded. Next, you invoke the `mdb` utility on the most recent `unix.n` and `vmcore.n` files.

Using the `::status` command, you examine the general status information about the dump. The next step is to display the console output by using the `::msgbuf` command.

Case Study 1: Walkthrough

```

...
<panic[cpu0]/thread=30002c57b00:
BAD TRAP: type=10 rp=2a100bb3500 addr=1397814 mmu_fsr=0
nm:
illegal instruction fault:
...
syncing file systems...
done
dumping to /dev/zvol/dsk/rpool/dump, offset 65536, content: kernel
> $r
%g0 = 0x0000000000000000          %l0 = 0x00000000018c2ca0 audit_active
%g1 = 0x00000300000b7540          %l1 = 0x0000030001f329c0
%g2 = 0x0000000000000240          %l2 = 0x000003000472d028
%g3 = 0x00000000018c9400          devimpl %l3 = 0x000002a100bb3658
%g4 = 0x0000000000000048          %l4 = 0x0000000000000400
%g5 = 0x0000000001397800          ksyms_open %l5 = 0x000003000472d0b0
%g6 = 0x0000000000000000          %l6 = 0x000003000472d0b8
%g7 = 0x0000030002c57b00          %l7 = 0x0000000000000440

%o0 = 0x0000000000000000          %i0 = 0x000002a100bb3758
%o1 = 0xffffffffffffffff          %i1 = 0x0000000000000201
%o2 = 0x000000000000001c          %i2 = 0x0000000000000002
%o3 = 0x0000000000000000          %i3 = 0x000003000036a008
%o4 = 0x000000000103c000          bzero+0x290 %i4 = 0x0000000070415ae0 ksyms_cb_ops
%o5 = 0x0000000000000000          %i5 = 0x0000000070415b68 ksyms_ops
%o6 = 0x000002a100bb2da1          %i6 = 0x000002a100bb2ea1
%o7 = 0x000000000139780c          ksyms_open+0xc %i7 = 0x000000000126ddc8
spec_open+0x4d0

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the output presented in the slide, note the value of %g7 (global register 7), and %pc (program counter), which is shown in the next slide. %g7 is the address of the thread running at the time of the panic on the CPU where the panic occurred. %pc is the address of the instruction that was executing when the panic occurred. Other panic types (for example, not BAD TRAP), might have other items that need examining besides the program counter—in particular, the stack backtrace, which will be discussed shortly.

Case Study 1: Walkthrough

```
<output continued from previous slide>

%ccr = 0x44 xcc=nZvc icc=nZvc
%fprs = 0x00 fef=0 du=0 dl=0
%asi = 0x80
%y = 0x0000000000000000
%pc = 0x0000000001397814 ksyms_open+0x14
%npc = 0x0000000001397818 ksyms_open+0x18
%sp = 0x000002a100bb2da1 unbiased=0x000002a100bb35a0
%fp = 0x000002a100bb2ea1

%tick = 0x0000000000000000
%tba = 0x0000000000000000
%tt = 0x10
%tl = 0x0
%pil = 0x0
%pstate = 0x016 cle=0 tle=0 mm=TSO red=0 pef=1 am=0 priv=1 ie=1 ag=0

%cwp = 0x00 %cansave = 0x00
%canrestore = 0x00 %otherwin = 0x00
%wstate = 0x00 %cleanwin = 0x00
. . . . .
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 1: Walkthrough

```

> <pc::dis
ksyms_mapin+0xd0:      ret
ksyms_mapin+0xd4:      restore    %g0, %i4, %o0
ksyms_mapin+0xd8:      mov        %i3, %i4
ksyms_mapin+0xdc:      ret
ksyms_mapin+0xe0:      restore    %g0, %i4, %o0
ksyms_open:           save       %sp, -0x100, %sp
ksyms_open+4:          mov        0x38, %o1
ksyms_open+8:          add        %fp, 0x7b7, %l3
ksyms_open+0xc:        call       -0x35ba9c    <bzero>
ksyms_open+0x10:       add        %fp, 0x7b7, %o0
ksyms_open+0x14:       illtrap    0x0
ksyms_open+0x18:       clr        %o2
ksyms_open+0x1c:       clr        %l6
ksyms_open+0x20:       ldx        [%i4 + 0xc8], %o1
ksyms_open+0x24:       sethi      %hi(0x1397400), %i4
ksyms_open+0x28:       call       -0x283d0c    <list_create>
ksyms_open+0x2c:       mov        %l3, %o0
ksyms_open+0x30:       mov        %l6, %o1
ksyms_open+0x34:       mov        %l6, %o2
ksyms_open+0x38:       add        %i4, 0xd0, %l4
ksyms_open+0x3c:       call       -0x286e08    <ksyms_snapshot>
>

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The system panicked because it tried to execute an `illtrap` instruction at `ksyms_open+0x14`. If you see a panic like this, it most likely denotes a serious hardware problem. The kernel will not overwrite its own code (due to permissions on the pages of memory containing kernel code). The other possibility is that the kernel branched to a location that contained the instruction `NULL`. In this case, look at the stack trace and the assembler code to see how the kernel got to the bad location. (That is a topic for a crash dump analysis course.)

Case Study 1: Walkthrough

```
> *panic_thread::print -t kthread_t
{
    struct _kthread *t_link = 0
    caddr_t t_stk = 0x2a100bb3ae0
    Lots of output omitted...
    klwp_t *t_lwp = 0x30006186078
    struct proc *t_procp = 0x300069c90c0
    Lots more output omitted...
}
> *panic_thread::print -t kthread_t t_procp
struct proc *t_procp = 0x300069c90c0
>
> *panic_thread::print -t kthread_t t_procp | ::print proc_t p_user.u_psargs
p_user.u_psargs = [ "/usr/ccs/bin/nm /dev/ksyms" ]
> $c
ksyms_open+0x14(2a100bb3758, 2001, 2, 3000036a008, 70415ae0, 70415b68)
spec_open+0x4d0(2a100bb3930, 100000, 3000036a008, 201, 3000472d028, 0)
fop_open+0x78(2a100bb3930, 2, 3000036a008, 2001, 3000772e680, 3000772e680)
vn_openat+0x500(0, 0, 1, 0, 2001, 7fffffff)
copen+0x260(fffffffffd19553, ffffffff7fffff30, 0, 10203, 0, 2001)
syscall_trap+0xac(fffffff7fffff30, 0, 10203, 0, 0, 0)
>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The next step is to examine the thread that was running at the time of the panic. To do this, use the `panic_thread` value (which is a pointer to the thread that was running when the panic occurred). You could also use global register 7 (`<g7::print kthread_t`).

Generally, the most useful information in the `kthread_t` is the pointer to the `proc_t` for the process in which the thread is running. Depending on the nature of the panic, any of the other fields might also be interesting. Use the `t_procp` member to get to the command-line arguments in effect at the time of the panic.

A user ran `/usr/ccs/bin/nm /dev/ksyms`, and the system panicked. Does this mean that user processes can cause panics? Only if there is a bug in the kernel, or if you do something incorrect with `mdb`, such as overwrite a valid instruction with garbage. One reason that you might want to know the command-line arguments that were running at the time of the panic is to determine if the panic is reproducible. To do this, try running the preceding command with the argument. It shouldn't result in a panic. You now look at the stack backtrace.

The `/usr/ccs/bin/nm` application was executing an `open(2)` system call, got to `ksyms_open+0x14`, and the machine panicked.

Case Study 1: Walkthrough

```

> ::cpuinfo -v
ID ADDR          FLG NRUN BSPL PRI RNRN KNRN SWITCH THREAD      PROC
0 0000183bb88 1b      1    0 49  no   no t-1  30002c57b00 nm

      RUNNING <--+ +--+> PRI THREAD      PROC
      READY      60 2a101103ca0 sched
      EXISTS
      ENABLE
...
> ::guit
# beadm list
BE          Active Mountpoint  Space  Policy Created
--          -
s11.rollback -          -      9.89G  static 2013-04-06 13:42
solaris     NR          /      51.64M static 2013-02-11 15:17
solaris.orig -          -      1.17M  static 2013-02-04 17:23
# beadm activate s11.rollback
# reboot
...

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You now look at what the CPUs were doing on the machine. In the example, the panic is artificial, and there is only one CPU on the machine. The output presented in the slide can be more useful on a multiprocessor system, especially when dealing with hangs.

Finally, check to see if an alternate boot environment (BE) exists for quick system recovery. Fortunately, the administrator created a backup BE (`s11.rollback` in this case) before performing the activity that caused the system panic. To restore the system to its pre-failure state, simply active the backup BE and reboot the system.

Practice 2 Overview

- 2-1: Using relevant log files
- 2-2: Using an application's debug mode
- 2-3: Applying troubleshooting techniques

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Identify the basic layers of an Oracle Solaris system
- Identify Oracle Solaris system error and fault types
- Describe troubleshooting techniques
- Gather information related to a system fault
- Mitigate the impact of system faults
- Analyze a case study that involves troubleshooting a system fault

ORACLE

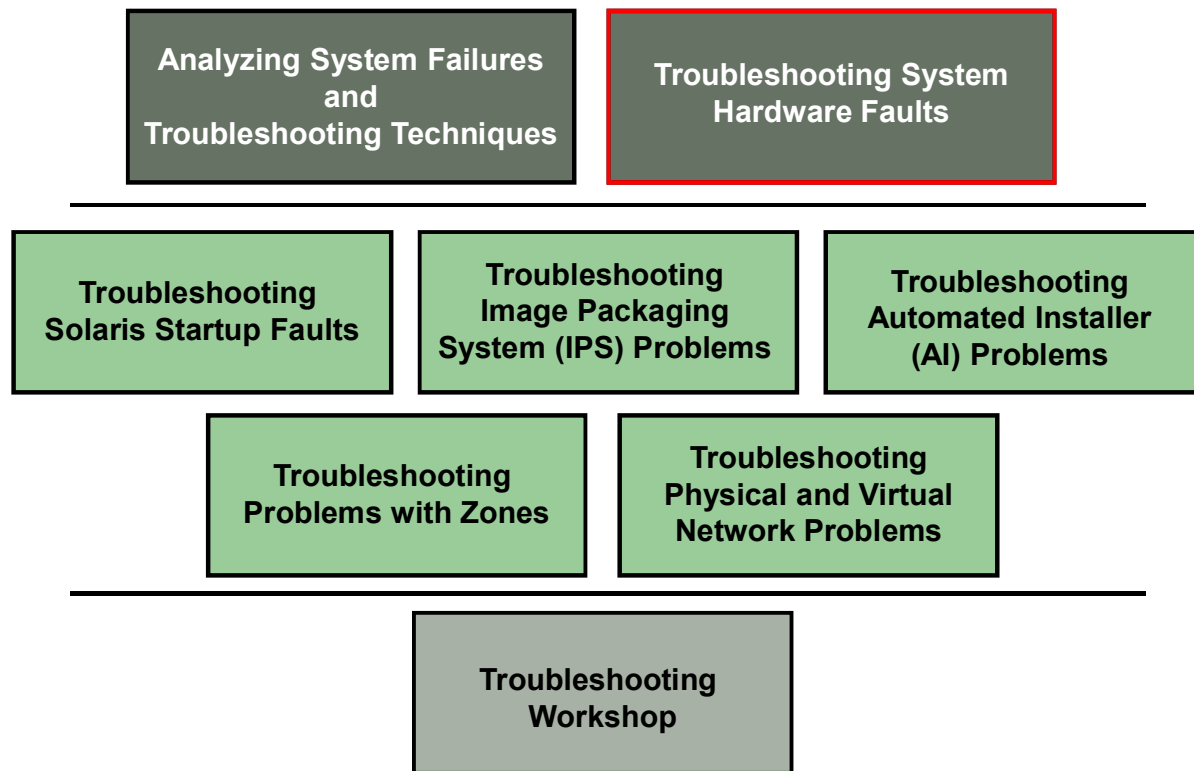
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Troubleshooting System Hardware Faults

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this lesson, “Troubleshooting System Hardware Faults”, you are introduced to a number of tools that you will use in this course and that you might find useful on the job for troubleshooting problems in Oracle Solaris 11. The tools include the SMF program, observability tools, such as the `stat` commands (for example, `dlstat`, `vmstat`, `flowstat`, and so on), `snoop`, `dtrace`, `truss`, and `ldd`, and the debugger tools `mdb` and `kmdb`.

Objectives

After completing this lesson, you should be able to:

- List components commonly found in Oracle enterprise-class systems
- Identify faulty hardware behavior
- Identify the best method for troubleshooting hardware faults
- Determine the best tools for analyzing and troubleshooting hardware faults
- Describe how to use the Server Processor (SP) when troubleshooting hardware faults
- Determine how to use the OpenBoot PROM when troubleshooting SPARC-related hardware faults
- Describe Solaris-based hardware diagnostics

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Identifying Oracle server hardware
- Troubleshooting hardware faults
- Service Processor and ILOM
- OpenBoot PROM
- POST
- Solaris-based hardware diagnostics
- Case Study 2

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Enterprise-Class Servers

- SPARC Servers
 - T4-series, T5-series, and M5
- SPARC SuperCluster
 - T4-4 System
- Sun Blade Servers
 - T4-1B Server, T5-1B Server, Sun Blade X3-2b Server
- Sun Netra Carrier-Grade Servers
 - Sun Netra Carrier Grade T4-series and Sun Netra 6000 Blade Servers
- Sun x86 Servers
 - X2 series, X3 series, Sun Blade X3-2b Server, and Netra Server X3 series
- Fujitsu M10 Servers
 - M10-1Server, M10-4 Server, and M10-4S Server

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

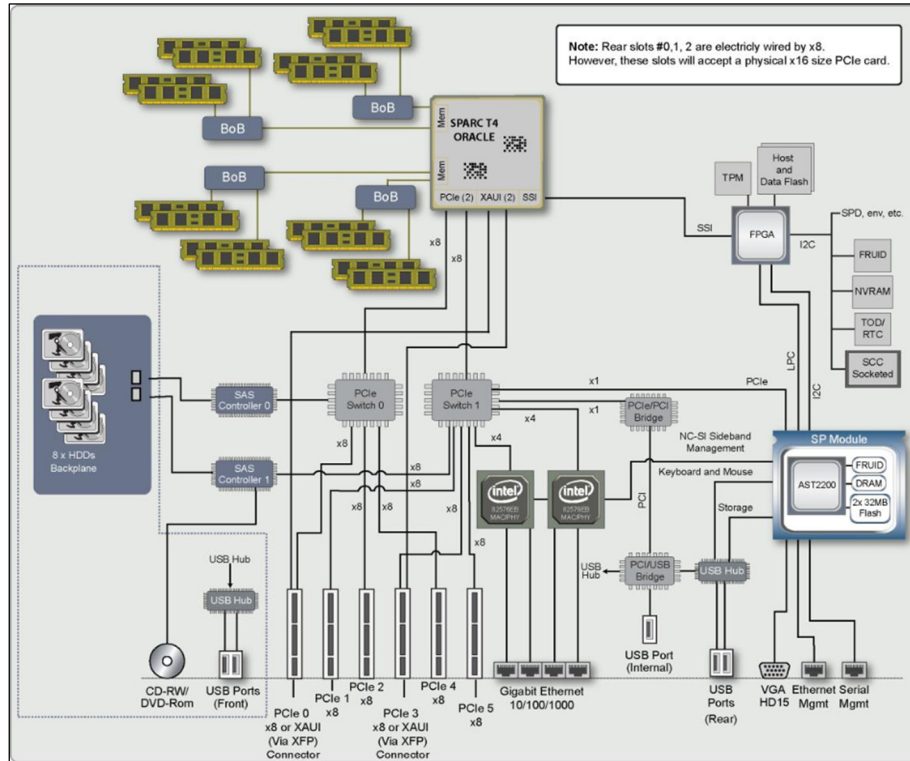
Oracle's servers are engineered to deliver record-breaking performance, simplified management, high availability, and cost-saving efficiencies. These industry-leading systems include built-in virtualization, cloud management and systems management, and are optimized to run Oracle Solaris, Oracle Linux, Oracle VM, and Oracle Enterprise Manager Ops Center. In addition, these servers support industry-leading software for Oracle and non-Oracle applications and solutions.

- **SPARC Servers:** Oracle's SPARC servers provide industry-leading price and performance along with the highest levels of availability and security required for mission-critical applications. The line of servers includes: SPARC T4-series, SPARC T5-series, and SPARC M5.
- **SPARC SuperCluster:** Oracle's SPARC SuperCluster T4-4 is a general-purpose engineered system that combines the computing power of the SPARC T4 processor, the scalability of Oracle Solaris 11, the database performance of Oracle Exadata storage, and the accelerated middleware processing of Oracle Exalogic Elastic Cloud.
- **Sun Blade Servers:** Oracle's Sun Blade modular systems integrate x86 and SPARC-based servers, storage, and advanced networking capabilities to support a complex and dynamic mix of IT workloads.

- **Sun Netra Carrier-Grade Servers:** Oracle offers the broadest portfolio of NEBS-certified servers designed for 4G infrastructures and cloud-based deployments, making it the safe choice for communications applications requiring continuous, uninterrupted service.
- **Sun x86 Servers:** Oracle's Sun x86 systems are the best platform for running Oracle software where x86 architectures are required. Only Oracle provides an optimized hardware and software stack that offers a choice of OS, virtualization software, and cloud management tools

It is important to note that each server model is made up of unique set of hardware components. When troubleshooting the server hardware, it is critical that you have some understanding of what major components make up the system. A good tool for identifying the major components that make up the system is the system block diagram.

System Block Diagram



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The system block diagram is a good tool for familiarizing yourself with the major hardware components that make up the system you are troubleshooting. This block diagram shows you the components found in the Oracle SPARC T4-2 server and how they are interconnected. Note that the major components include: CPUs, memory, I/O buses, internal devices, and service processor (SP). Block diagrams are normally provided in the server hardware installation manual.

Agenda

- Identifying Oracle server hardware
- Troubleshooting hardware faults
- Service Processor and ILOM
- OpenBoot PROM
- POST
- Solaris-based hardware diagnostics
- Case Study 2

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Hardware Fault Behavior

- Faulty power supplies and fans:
 - Power-on or system startup failures or lockups
 - Spontaneous rebooting or intermittent lockups during normal operation
- Faulty memory module:
 - System panics
 - Erratic application or OS behavior
 - Application core dumps
- Faulty CPU:
 - System panics
 - Erratic application or OS behavior
 - Application core dumps
- Faulty disk drive
 - Power-on self-test (POST) is okay but system fails to boot.
 - System I/O operations hang.
 - Data corruption

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the slide are some common behaviors caused by server hardware failures.

Diagnosing Hardware Faults

- LEDs
- ILOM
- Power-on self-test (POST)
- Oracle Solaris OS Predictive Self-Healing (PSH)
- Solaris OS logs
- Oracle SunVTS

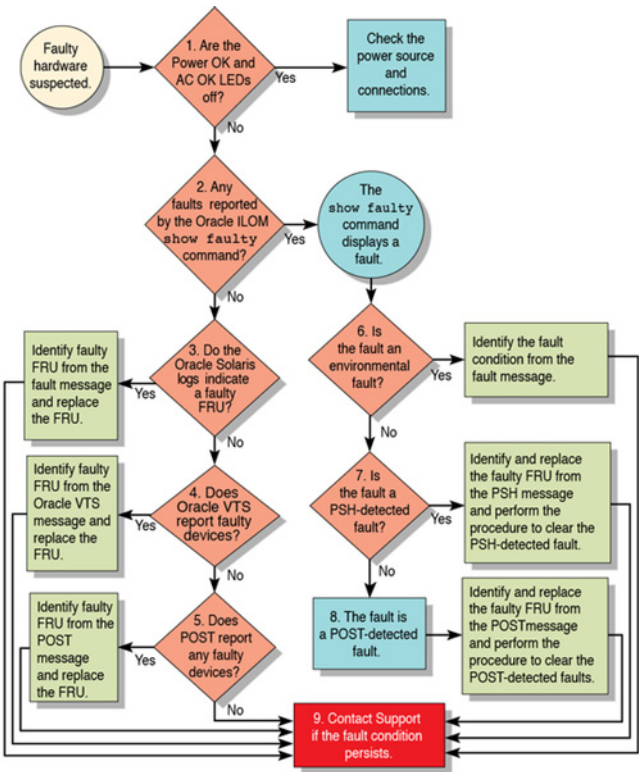
The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle servers provides a range of system management and diagnostic tools that enable you to monitor server operations and troubleshoot server problems. The following is a high-level summary of the various diagnostic tools that are available on the server:

- **LEDs:** Status indicators are provided on the server's front and rear panels as well as on some FRUs.
- **ILOM:** The Oracle Integrated Lights Out Manager (ILOM) firmware runs on the service processor and is the central software resource for identifying and managing server problems.
- **Power-on self-test (POST):** POST performs diagnostics on system components whenever the system is turned on or a system reset is issued.
- **Oracle Solaris OS Predictive Self-Healing (PSH):** PSH continuously monitors the health of the processor and memory, collecting data that can be used to predict component failures before they occur. PSH works with ILOM to take a component offline if data indicates there is a high risk that it will fail.
- **Solaris OS log files and command interface:** The Solaris OS reports operational and error information to its standard log files. It also provides various commands for investigating system status.
- **Oracle SunVTS:** This is an application that exercises the system, provides hardware validation, and discloses possible faulty components with recommendations for repair.

Hardware Troubleshooting Flowchart



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This flowchart illustrates the complementary relationship of the different diagnostic tools and indicates a default sequence of use.

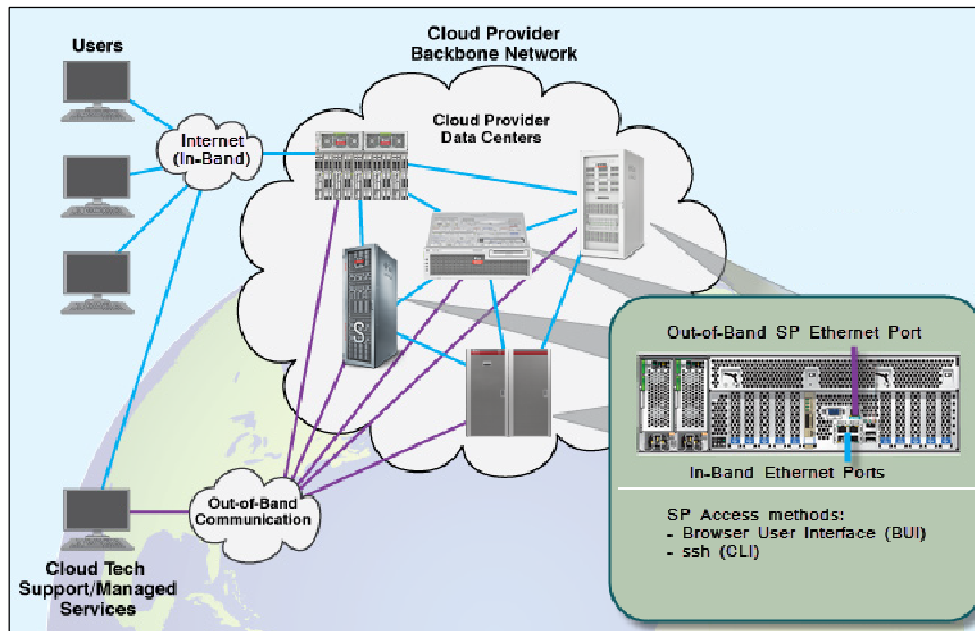
Agenda

- Identifying Oracle server hardware
- Troubleshooting hardware faults
- **Service Processor and ILOM**
- OpenBoot PROM
- POST
- Solaris-based hardware diagnostics
- Case Study 2

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Service Processor (SP)

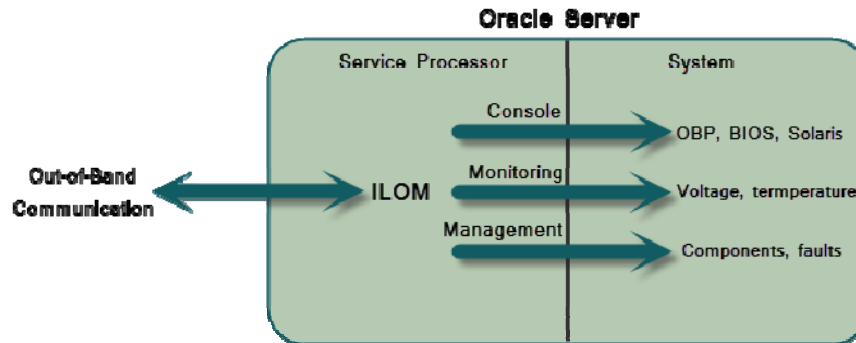


ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A service processor (SP) is a separate dedicated internal processor located on the motherboard of a server, a PCI card, or on the chassis of an Oracle server or telecommunications platform. It operates independently from the server's CPUs, memory, I/O, and operating system (OS). The service processor monitors the server's on-board instrumentation (temperature sensors, CPU status, fan speed, voltages), provides remote reset or power-cycle capabilities, and enables remote out-of-band access to SPARC OpenBoot Prom or x86 BIOS configuration or OS console interface.

Integrated Lights Out Manager (ILOM)



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle's Integrated Lights Out Manager (ILOM) provides advanced service processor hardware and software that you can use to manage and monitor your Oracle Sun servers. Oracle ILOM's dedicated hardware and software is pre-installed on a variety of Oracle Sun server platforms, including x86-based Sun Fire servers, Sun Blade modular chassis systems, Sun Blade server modules, as well as on SPARC-based servers. Oracle ILOM is a vital management tool in the data center and can be used to integrate with other data center management tools already installed on your systems. Oracle ILOM is supported on many Oracle systems, enabling users to experience a single, consistent, and standards-based service processor (SP) across all Oracle Sun server product lines. This means you will have:

- Single, consistent system management interfaces for operators
- Rich protocol and standards support
- Broadening third-party management support
- System management functions integrated into Oracle's Sun servers

What Can You Do with ILOM?

- You can remotely manage your server as if you were using a locally attached keyboard, monitor, and mouse.
- Using ILOM, you can:
 - Learn about hardware errors and faults as they occur.
 - Remotely control the power state of your server.
 - View the graphical and non-graphical consoles for the host.
 - View the current status of sensors and indicators on the system.
 - Determine the hardware configuration of your system.
 - Receive generated alerts about system events in advance through IPMI PETs, SNMP traps, or email alerts.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

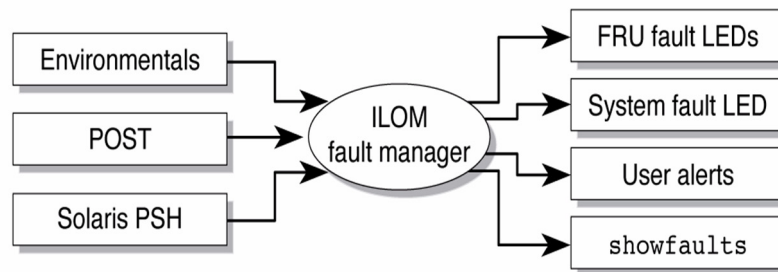
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle ILOM enables you to actively manage and monitor the server independently of the operating system state, providing you with a reliable lights-out management (LOM) system. The Oracle ILOM service processor (SP) runs its own embedded operating system and has a dedicated Ethernet port, which together provide out-of-band management capability. In addition, you can access Oracle ILOM from the server's host operating system. Using Oracle ILOM, you can remotely manage your server as if you were using a locally attached keyboard, monitor, and mouse. Oracle ILOM automatically initializes as soon as power is applied to your server. It provides a full-featured, browser-based web interface and has an equivalent command-line interface (CLI). There is also an industry-standard SNMP interface and IPMI interface. With Oracle ILOM, you can proactively:

- Learn about hardware errors and faults as they occur
- Remotely control the power state of your server
- View the graphical and non-graphical consoles for the host
- View the current status of sensors and indicators on the system
- Determine the hardware configuration of your system
- Receive generated alerts about system events in advance through Intelligent Platform Management Interface (IPMI) platform event traps (PETs), SNMP traps, or email alerts.

ILOM Fault Handling

Faults detected by ILOM, POST, and the Solaris Predictive Self-Healing (PSH) technology are forwarded to ILOM for fault handling.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ILOM enables you to remotely run diagnostics such as POST, that would otherwise require physical proximity to the server's serial port. You can also configure ILOM to send email alerts of hardware failures, hardware warnings, and other events related to the server or to ILOM.

The service processor runs independently of the server, using the server's standby power. Therefore, ILOM firmware and software continue to function when the server OS goes offline or when the server is powered off.

Faults detected by ILOM, POST, and the Solaris Predictive Self-Healing (PSH) technology are forwarded to ILOM for fault handling. In the event of a system fault, ILOM ensures that the Service Required LED is turned on, FRUID PROMs are updated, the fault is logged, and alerts are displayed. Faulty FRUs are identified in fault messages using the FRU name.

The service processor can detect when a fault is no longer present. When this happens, it clears the fault state in the FRU PROM and extinguishes the Service Required LED.

A fault condition can be removed in two ways:

- **Unaided recovery:** Faults caused by environmental conditions can clear automatically if the condition responsible for the fault improves over time.
- **Repaired fault:** When a fault is repaired by human intervention, such as a FRU replacement, the service processor will usually detect the repair automatically and extinguish the Service Required LED.

Entering and Exiting ILOM

- Log in to ILOM by using `ssh`.

```
$ ssh username@SP_ipaddress  
Password: password  
->
```

- The default password is `changeme`.

- Use `exit` to quit ILOM.

```
-> exit
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows you how to log in to and exit ILOM.

ILOM Troubleshooting Tasks

ILOM Task	ILOM Commands
Get help.	-> help -> help [command]
View host firmware version.	-> show /HOST sysfw_version
Open a system console.	-> start /SP/console
Change power state.	-> start /SYS -> stop [-f] /SYS
Reset SP and system.	-> reset /SP -> reset /SYS
View component information.	-> cd target -> show
View the ILOM event log.	-> cd /SP/logs/event -> show list
Displays information about the operating state of the host system.	-> show /HOST



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ILOM Troubleshooting Tasks

ILOM Task	ILOM Commands
Discover faults by using ILOM.	-> show /SP/faultmgmt
Enable POST diagnostics.	-> set /SYS/keyswitch_state=diag
View faulty components.	-> show faulty
Configure automatic system recovery (ASR).	-> set /HOST/diag mode=normal -> set /HOST/diag level=max -> set /HOST/diag trigger=power-on-reset -> start /SP/console ok setenv auto-boot true ok setenv auto-boot-on-error? true ok reset-all
Collect SP data to diagnose system problems.	-> set /SP/diag/snapshot dataset=full
Invoke the Solaris fault manager shell and view faults.	-> start /SP/faultmgmt/shell Are you sure you want to start /SP/faultmgmt/shell (y/n)? y faultmgmtsp> fmadm faulty



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table shows the components tree for an Oracle Sun SPARC Enterprise T5240 server. Note that the component tree is obtained from the system firmware (OpenBoot PROM or BIOS) during system startup after the completion of POST.

ILOM Fault Detection: Examples

- Fault detected by PSH:

```
-> show faulty
```

Target	Property	Value
/SP/faultmgmt/0	fru	/SYS/MB/CMP0/BR0/CH1/D0
/SP/faultmgmt/0	timestamp	Dec 14 22:43:59
/SP/faultmgmt/0/ faults/0	sunw-msg-id	SUN4V-8000-DX
/SP/faultmgmt/0/ faults/0	uuid	3aa7c854-9667-e176-efe5-e487e520 7a8a
/SP/faultmgmt/0/ faults/0	timestamp	Dec 14 22:43:59

- Fault detected by POST:

```
-> show faulty
```

Target	Property	Value
/SP/faultmgmt/0	fru	/SYS/MB/CMP0/BR1/CH0/D0
/SP/faultmgmt/0	timestamp	Dec 14 23:01:32
/SP/faultmgmt/0/ faults/0	timestamp	Dec 14 23:01:32 faults/0
/SP/faultmgmt/0/ faults/0	sp_detected_fault	/SYS/MB/CMP0/BR1/CH0/D0 Forced fail (POST)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows two examples of using the ILOM show faulty command to identify a failed hardware component. The first example shows a faulty memory module (/SYS/MB/CMP0/BR1/CH0/D0) detected by PSH. The second example shows the same memory module fault detected during POST.

Automatic System Recovery (ASR)

- ASR enables the system to resume operation after experiencing certain nonfatal hardware faults.
 - Nonfatal hardware faults are associated with memory modules and PCI cards.
- You apply ASR to firmware-based diagnostics by using ILOM and OBP.

```
-> set /HOST/diag mode=normal
-> set /HOST/diag level=max
-> set /HOST/diag trigger=power-on-reset
-> start /SP/console
ok setenv auto-boot true
ok setenv auto-boot-on-error? true
ok reset-all
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The system provides for Automatic System Recovery (ASR) from failures in memory modules or PCI cards.

Automatic System Recovery functionality enables the system to resume operation after experiencing certain nonfatal hardware faults or failures. When ASR is enabled, the system's firmware diagnostics automatically detect failed hardware components. An autoconfiguring capability included in the system firmware enables the system to unconfigure failed components and to restore system operation. As long as the system is capable of operating without the failed component, the ASR features enable the system to reboot automatically, without operator intervention.

Agenda

- Identifying Oracle server hardware
- Troubleshooting hardware faults
- Service Processor and ILOM
- **OpenBoot PROM**
- POST
- Solaris-based hardware diagnostics
- Case Study 2

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Boot PROM for SPARC Systems

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

All Oracle Solaris SPARC systems have resident boot PROM firmware that controls the operation of the system before the operating system has been booted. The firmware within Sun systems that are based on SPARC technology is known as OpenBoot firmware. In addition to booting the operating system, the OpenBoot architecture enables the boot PROM firmware to do the following:

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

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

Note: The boot PROM is not used to modify the Solaris OS file systems or files. It handles mainly hardware devices. The OS works with and is dependent on firmware, but firmware is independent of the OS.

Boot PROM features

OBP Feature	Description
FCode interpreter	Enables using the same device driver on systems with different CPU instruction sets
POST	Performs a series of diagnostic tests on the hardware components to verify that they are functioning
Plug-in device drivers	Enables the adding and removing of various peripheral cards without making any changes to the boot PROM
Programmable user interface	Enables the combining of user commands to make complete programs for debugging hardware and software
Facilities for dynamically constructing a device tree structure in nonpageable memory	Probes hardware devices and dynamically constructs a device tree data structure in nonpageable memory
Diagnostic commands	Help to perform system reconfiguration and hardware diagnostic tasks

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

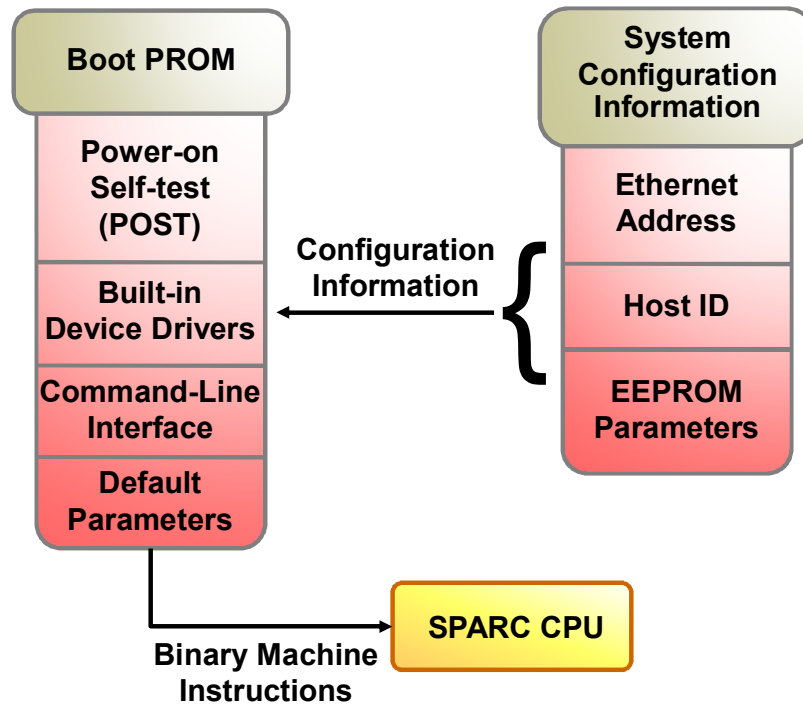
Information about the boot PROM features helps to define the capabilities of OBP and the scope of activities that can be performed using the OBP diagnostic commands. The table presented in the slide shows the main features of boot PROM.

Additional information on each feature follows:

- **FCode interpreter:** Most plug-in drivers are written in a machine-independent language called FCode. The OBP includes an FCode interpreter that enables using the same device driver on systems with different CPU instruction sets.
- **POST:** POST performs a series of diagnostic tests on the hardware components of the system board to verify that all the components are functioning.
- **Plug-in device drivers:** Every SBus and peripheral component interconnect (PCI) card contains a firmware chip called an IDPROM that has a minimal device driver and a customized POST routine. The IDPROM enables the peripheral card to specify how to test and probe the system at the firmware level. This enables adding and removing various peripheral cards without making any changes to the boot PROM.
- **Programmable user interface:** OBP is based on the interactive Forth language. This enables combining user commands to make complete programs for debugging hardware and software.

- **Facilities for dynamically constructing a device tree structure in nonpageable memory:** The OBP probes hardware devices and dynamically constructs a device tree data structure in nonpageable memory. The device tree is organized as a hierarchy and represents all the hardware devices available on the system. The device tree can be viewed at the firmware level to determine which hardware is available on the system.
- **Diagnostic commands:** OBP provides an extensive set of diagnostic commands that help to perform system reconfiguration and hardware diagnostic tasks.

Boot PROM Initialization Sequence



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

Note: When the boot process has completed and the operating system is running, you see a login prompt displayed on the console. When the operating system is not running, the `ok` prompt is displayed.

Checking the OBP Firmware Revision

To check the OBP firmware revision on the system, use one of the following commands:

- The `banner` or `.version` command at the `ok` prompt

```
ok banner
Sun Fire T200, No Keyboard
Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #67263974.
Ethernet address 0:14:4f:2:5d:e6, Host ID: 84025de6.
```

- The `prtconf -V` command at the Shell prompt

```
# prtconf -V
OBP 4.30.4.d 2011/07/06 14:29
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can check the OBP firmware revision on a system by using either the `banner` or `.version` command at the `ok` prompt or the `prtconf -V` command at the Shell prompt.

The `banner` command displays hardware configuration information, including the revision number of the PROM, as shown in the first example presented in the slide. The `.version` command displays the version and date of the startup PROM, as shown in the following example:

```
Release 4.30.4.d created 2011/07/06 14:29
```

The `prtconf -V` command displays platform-dependent PROM on SPARC platforms, as shown in the second example in the slide.

Identifying the OBP Variables Used in Troubleshooting

OBP Variable	Description
auto-boot?	Specifies whether a system boots automatically after a power-on or reset of the system
diag-device	Specifies the diagnostic boot source device
diag-switch?	Specifies the diagnostic mode in which the system runs
pcia-probe-list	Controls the probe order of the plug-in devices attached to the <code>pcia</code> bus
pcib-probe-list	Controls the probe order of the plug-in devices attached to the <code>pcib</code> bus
security-mode	Controls the firmware security level
watchdog-reboot?	Specifies whether the system must reboot automatically when a watchdog reset occurs



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use various OBP commands to query and set OBP variables. OBP variables provide the flexibility to modify the default behavior of different aspects of the OBP firmware.

The table in the slide displays some common OBP variables along with their descriptions.

Displaying OBP Variables

To print the OBP variables along with their values, use the `printenv` command at the `ok` prompt.

<code>ok printenv</code>		
Variable Name	Value	Default Value
<code>ttya-rts-dtr-off</code>	<code>false</code>	<code>false</code>
<code>ttya-ignore-cd</code>	<code>true</code>	<code>true</code>
<code>keyboard-layout</code>		
<code>reboot-command</code>		
<code>security-mode</code>	<code>none</code>	No default
<code>security-password</code>		No default
<code>security-#badlogins</code>	<code>0</code>	No default
<code>verbosity</code>	<code>min</code>	<code>min</code>
<code>pci-mem64?</code>	<code>false</code>	<code>false</code>
<code>diag-switch?</code>	<code>false</code>	<code>false</code>
<code>local-mac-address?</code>	<code>true</code>	<code>true</code>
<code>fcode-debug?</code>	<code>false</code>	<code>false</code>
<code>scsi-initiator-id</code>	<code>7</code>	<code>7</code>
<output omitted>		

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To view the OBP variables on a system, you can use the `printenv` command.

Note: If you run the `printenv` command on the system, all the OBP variables with their corresponding default values are displayed.

A partial output of the command on a Sun Fire T2000 is presented in the example in the slide.

You can also display a single OBP variable and its value by using the `printenv` command with the variable name. For example, to display the value of the `boot-device` variable, type the following at the `ok` prompt:

```
ok printenv boot-device
```

```
boot-device = /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a
disk net
```

Using the `setenv` and `set-default` Commands

- To set an OBP parameter, use the `setenv` command.
- To reset a parameter back to the default value, use the `set-default` command:

```
ok setenv auto-boot? false
auto-boot? =      false
ok set-default auto-boot?
ok printenv auto-boot?
auto-boot? =      true
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

As shown by the `printenv` output in the previous slide, the `auto-boot` parameter's default is `true`. To change the setting, use the `setenv` command. The `set-default` command resets a single OBP variable to its default value. The `printenv` command can also be used to show one variable as shown in this slide.

Using the `eeeprom` Command

The `eeeprom` command:

- Is used to view and modify OBP variables from the shell
- Stores OBP variable changes in the NVRAM chip
- Does not reset the system

Syntax	Use
# <code>eeeprom</code>	Displays all the variables with their corresponding values
# <code>eeeprom parameter</code>	Displays a single parameter and its current value
# <code>eeeprom parameter=value</code>	Changes the current value of a parameter

```
# eeeprom 'auto-boot?'=false
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `eeeprom` command is used to view and modify OBP variables from the shell. However, superuser access rights are required to modify any OBP variable with the `eeeprom` command. The `eeeprom` command stores the changes made to the OBP variables in the NVRAM chip without resetting the system.

The table in the slide shows various uses of the `eeeprom` command and the corresponding syntax.

Note: When using the `eeeprom` command on variables that have a question mark, either enclose the variable in quotation marks or precede the question mark with an escape character (`\`). This prevents the shell from interpreting the question mark.

In the example presented in the slide, the `eeeprom` command is used to prevent the system from automatically booting after completing the POST diagnostics by altering the `auto-boot?` variable.

Using Manual OBP Diagnostic Commands

This section covers the following topics:

- Using the `probe` commands
- Using the `test` commands
- Using the `watch` commands

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this section, you are shown the commands used to determine the status of hardware components at the firmware level. All these commands are run from the `ok` prompt.

Using the probe Commands

To diagnose peripheral devices, use the `probe` commands.

Command	Purpose	Output
<code>probe-ide</code>	Probes the Integrated Drive Electronics (IDE) devices that are connected to the on-board IDE interface of the system	Displays the target address, unit number, device type, and manufacturer name of each IDE device
<code>probe-scsi</code>	Probes the small computer system interface (SCSI) devices, such as disks and tape drives, that are attached to the on-board SCSI controller	Displays the target address, unit number, device type, and manufacturer name of each SCSI device
<code>probe-scsi-all</code>	Probes the devices attached to the on-board SCSI controller as well as the devices that are attached to the PCI and SBus SCSI controllers	Identifies SCSI devices by their target addresses

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The probe commands, as shown in the table in the slide, are used to diagnose peripheral devices, such as disks, tape drives, and CD-ROMs that are connected to the system.

If the system has a SCSI card attached to it, use the `probe-scsi-all` command. The `probe-scsi` command probes only the on-board SCSI controllers and not the peripheral card controllers. Before you use the `probe-scsi` or `probe-scsi-all` commands, power-on all the SCSI devices. This is necessary because the `probe-scsi` command can detect the connected SCSI devices only if the devices are powered on.

Using the test Commands

To test the hardware devices attached to the system, use the `test` commands.

Command	Purpose
<code>test-all</code>	Tests all the devices in a system, such as the SBus cards that have a built-in test program. This command does not test the tape drives, DVDs, and hard disks.
<code>test-net</code>	Performs internal and external loopback tests on the auto-selected system Ethernet interface

Caution: Before running the `test` commands, reset the system once after dropping to the `ok` prompt. This helps to clear all buffers and registers and ensures that the system does not hang.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `test` commands, as shown in the table in the slide, test the hardware devices attached to the system.

Using the `watch` Commands

To monitor the network traffic and clock function of the system, use the `watch` commands.

Command	Purpose
<code>watch-net</code>	Monitors broadcast Ethernet packets on the Ethernet cables that are connected to the system
<code>watch-net-all</code>	Monitors Ethernet packets on all the Ethernet interfaces that are installed on the system
<code>watch-clock</code>	Displays seconds from the TOD chip in NVRAM

Caution: Before running the `watch` commands, reset the system once after dropping to the `ok` prompt. This helps to clear all buffers and registers and ensures that the system does not hang.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `watch` commands, as shown in the table in the slide, monitor the network traffic and clock function of the system.

Quiz

Which command can you use to set the appropriate variable so that the system does not boot automatically? (Select two.)

- a. `ok setenv auto-boot? false`
- b. `ok printenv auto-boot?`
- c. `# eeprom 'auto-boot?'=false`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, c

Note: You can also set this by using the boot control commands in ILOM/ALOM. For more information, see the appendix included in this course.

Quiz

Which command can you use to display the list of all OBP variables on a system? (Select two.)

- a. # eeprom
- b. ok printenv
- c. ok setenv

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, b

Agenda

- Identifying Oracle server hardware
- Troubleshooting hardware faults
- Service Processor and ILOM
- OpenBoot PROM
- **POST**
- Solaris-based hardware diagnostics
- Case Study 2

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Introducing POST

This section covers the following topics:

- Identifying POST components
- Enabling extended POST diagnostics
- Examining and manipulating the OBP device tree
- Viewing extended diagnostics during POST

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Identifying POST Components

Use POST to perform diagnostic tests for the following hardware components:

- CPU modules
- Memory management units (MMUs)
- Memory, such as the `Init Memory` and `Block Memory Addr Tests` parameters
- Interrupts
- NVRAM, such as `NVRAM Battery Detect Test`, `NVRAM Scratch Addr Test`, and `NVRAM Scratch Data Test` parameters
- Cache, such as `Ecache Tests` and `Basic Cache Tests` parameters
- Register tests

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

POST is a binary program written for the SPARC processor. It is a stand-alone piece of firmware that does not depend on OBP. You can use POST to initialize and test the hardware that is part of the system. POST performs a series of diagnostic tests on the hardware components of the system board to verify that all the components are functioning properly. POST also helps to determine which components failed and must be replaced. The error messages displayed during the POST sequence help administrators and support personnel to determine if hardware problems exist on the system.

Note: POST does not perform extensive tests on any components of the main logic board, such as SBus or PCI cards and associated I/O devices. These tests are performed by using OBP.

Enabling Extended POST Diagnostics

This section covers the following topics:

- Setting the `diag-switch?` variable
- Setting the `diag-level` variable

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To implement extended POST diagnostic tests, you must enable extended POST on the Solaris firmware. Extended POST diagnostic tests are invoked automatically when the system is powered on, if the following conditions have been met:

- The `diag-switch?` variable is set to `true`.
- The value of the `diag-level` variable is set to `max`.
- You pressed the Stop + D keys after powering on the system.

Note: You must also ensure that the `diag-device` parameter is set to a bootable device when extended POST is enabled. The `diag-device` parameter is described later in this lesson.

In this section, you are shown how to set each of the variables.

Setting the diag-switch? Variable

Use the `diag-switch?` variable to control whether POST is executed in normal or in extended diagnostic mode.

Mode	Value	Description
Normal diagnostic	Defaults to <code>false</code> on most desktops	In this mode, no progressive test messages are displayed on the console during the execution of POST tests. If an error occurs, the error messages are displayed on either the tty-type terminal or on the console.
Extended diagnostic	Defaults to <code>true</code> on most servers	Displays progressive test messages on the console during execution. If POST is successful, the control is transferred to OBP. OBP then probes the installed <i>peripheral</i> bus modules.

To set the value of the `diag-switch?` variable to `true`, run the `setenv` command at the `ok` prompt:

```
ok setenv diag-switch? true
ok
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `diag-switch?` variable is an OBP variable that controls whether POST is executed in normal diagnostics mode or in extended diagnostics mode. When the value of the `diag-switch?` variable is set to `false`, POST is executed in normal diagnostic mode. When the value of the `diag-switch?` variable is set to `true`, POST is executed in extended diagnostic mode. The normal and extended modes are described in the table presented in the slide.

When the `diag-switch?` variable is set to `true`, the system:

- Performs self-tests during any subsequent system power-on
- Displays additional status messages.
- Uses different configuration variables to boot a system, for example:
 - If the `auto-boot?` variable is set to `true`, the system boots from the boot device specified in the `diag-device` variable.
 - If the `auto-boot?` variable is set to `false`, the system remains at the PROM monitor without booting.

Setting the `diag-level` Variable

You can select three different levels of diagnostic tests for POST:

- **`off` (no testing):** No tests are performed during system power-on.
- **`max` (maximum):** POST performs an extended set of diagnostic-level tests.
- **`min` (minimum):** POST performs an abbreviated set of diagnostic-level tests.

To perform extended diagnostic tests, run the `setenv` command at the `ok` prompt.

```
ok setenv diag-level max
ok
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are three different levels of diagnostic tests for POST. You can choose to run all or some of the tests. The OBP firmware initiates the selected level of POST based on the value you have set for the `diag-level` variable.

Using the `show-post-results` Command

To view the last executed POST:

1. Set the `diag-switch?` variable to `true`.
`ok setenv diag-switch? true`
2. Set the `diag-level` variable to `max`.
`ok setenv diag-level max`
3. Set the `auto-boot?` variable to `false`.
`ok setenv auto-boot? false`
4. Power down the system.
5. Power on the system.
6. Run `show-post-results` to view the POST output.
`ok show-post-results`
<output omitted>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `show-post-results` command is an OBP-level command that you use at the firmware level of a system to display information about the last executed POST.

Note: If you have a terminal device, you can view the POST output in real time by attaching the terminal device to a serial port. If you do not have a terminal device, you can use the `show-post-results` command to view the results after executing POST diagnostics.

When POST has not been executed, the `show-post-results` command simply returns the following message at the `ok` prompt: Power On Selftest not run on last reset.

OBP Device Tree: Overview

- Solaris hardware uses a device tree to organize the devices that are attached to the system.
- OpenBoot firmware:
 - Builds the device tree from the information generated during POST
 - Loads the device tree into memory
- You can examine the device path by using the:
 - `/devices` directory
 - `prtconf -vp` command

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Solaris hardware uses the concept of a device tree to organize devices that are attached to the system. In SPARC-based systems, the OpenBoot firmware builds a device tree from the information generated during POST and loads the device tree into memory.

The kernel refers to the device tree during the boot process to determine the hardware configuration of the system. This information is used to identify the card and slot configuration on the system and to map the driver names, unit addresses, and device arguments to the physical devices and their locations on the system.

The device path on a system can be examined by using the following:

- **`/devices` directory:** This directory contains the physical device names for the devices that are attached to the system.
- **`prtconf -vp` command:** The `-v` specifies verbose mode. The `-p` displays information derived from the device tree provided by the firmware (PROM) on SPARC platforms or the booting system on x86 platforms. The device tree information displayed by using this option is a snapshot of the initial configuration and may not accurately reflect reconfiguration events that occur later.

This table shows the OBP device identifiers for an Oracle Sun SPARC Enterprise T5240 server. Note that the OBP device tree is unique to each server model because the device hardware is different for each server model.

Examining Device Path Naming

Format for the name of a device tree node:

`device-name@unit-address:device-arguments`

Path Name Parameter	Description
<i>device-name</i>	Includes the manufacturer and model names of devices, separated by a comma
@	Precedes the address parameter
<i>unit-address</i>	Includes a text string that stores the physical address of the device in the address space of the parent node
: (colon)	Precedes the device argument parameter
<i>device-arguments</i>	Includes a text string that is used to pass additional information to the device software

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A system identifies devices by the path name of the device nodes. A node in the device tree represents a device or a firmware service. Many nodes include subnodes that usually represent system buses and controllers associated with system buses. The subnodes represent the devices that are connected to buses or controllers.

Each device has a unique path name that represents the type of device and the location of the device in the overall addressing structure. In a device tree, the path for a device begins with a slash (/). The slash represents the root of the device tree. The table in the slide presents the parameters of a device path name. Additional information about some of the parameters follows:

- ***device-name***: This parameter includes a string of 1 to 31 characters, such as punctuation characters, which have mnemonic values. This parameter in a device path name is case-sensitive.
- ***unit-address***: The format of this parameter is bus-dependent.
- ***device-arguments***: The format of this parameter is device-dependent.

Note: The path of a device in a device tree varies, depending on the type of system and the configuration of the device.

Solaris path_to_inst File

```
# cat /etc/path_to_inst
...
"/pci@780/pci@0/pci@1" 1 "pcieb"
"/pci@780/pci@0/pci@1/network@0" 0 "e1000g"
"/pci@780/pci@0/pci@1/network@0,1" 1 "e1000g"
...
"/pci@7c0/pci@0/pci@1/pci@0/isa@2" 0 "ebus"
"/pci@7c0/pci@0/pci@1/pci@0/isa@2/serial@0,3f8" 0 "su"
"/pci@7c0/pci@0/pci@1/pci@0/usb@5" 0 "ohci"
"/pci@7c0/pci@0/pci@1/pci@0/usb@6" 1 "ohci"
...
"/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2" 0 "mpt"
"/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@0,0" 2 "sd"
"/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@1,0" 3 "sd"
"/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@2,0" 4 "sd"
...
"/pci@7c0/pci@0/pci@2/network@0" 2 "e1000g"
"/pci@7c0/pci@0/pci@2/network@0,1" 3 "e1000g"
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When the Solaris OS is installed, OBP passes its device tree (devices that it is aware of) to the OS during the initial boot. The devices are then enumerated. The Solaris `/etc/path_to_inst` file records mappings of physical device names to instance numbers. The instance number of a device is encoded in its minor number, and is the way in which a device driver determines which of the possible devices that it may drive is referred to by a given special file. To keep instance numbers persistent across reboots, the system records them in `/etc/path_to_inst`. The slide example shows a portion of the `path_to_inst` file from an Oracle Sun Fire T2000 server.

Listing Device Tree Information

This section covers the following topics:

- Using the `show-devs` command
- Using the `show-disks` command
- Using the `show-nets` command

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Using the show-devs Command

Use the `show-devs` command at the `ok` prompt to view the entire device tree.

```
ok show-devs
/os-io
/ramdisk-root
/pci@7c0
/pci@780
/cpu@17
/cpu@16
/cpu@15
/cpu@14
/cpu@13
/cpu@12
/cpu@11
/cpu@10
/cpu@f
/cpu@e
<output omitted>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `show-devs` command displays the entire device tree. With this command, you can view a list of all the device tree paths that are available from the `root` level.

The example in the slide shows the type of output that is generated when you run the `show-devs` command.

Note: To display all the devices directly under a specific device in the device tree, use the `show-devs` command with the device path, as shown in the following example.

```
ok ok show-devs /pci@7c0
/pci@7c0/pci@0
/pci@7c0/pci@0/pci@9
/pci@7c0/pci@0/pci@8
/pci@7c0/pci@0/pci@2
/pci@7c0/pci@0/pci@1
/pci@7c0/pci@0/pci@2/network@0,1
/pci@7c0/pci@0/pci@2/network@0
<output omitted>
```

Using the `show-disks` Command

Use the `show-disks` command at the `ok` prompt to:

- Display the available disks on the system
- Select the device path that relates to the disk that is used in a custom device alias

```
ok show-disks
a) /ramdisk-root
b) /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk
c) /pci@7c0/pci@0/pci@1/pci@0/ide@8/cdrom
d) /pci@7c0/pci@0/pci@1/pci@0/ide@8/disk
q) NO SELECTION
Enter Selection, q to quit: d
/pci@7c0/pci@0/pci@1/pci@0/ide@8/disk has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
        for creating devalias mydev for
        /pci@7c0/pci@0/pci@1/pci@0/ide@8/disk
ok
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `show-disks` command displays the available disks on the system and enables you to select the device path that relates to the disk that is used in a custom device alias.

The output presented in the example in the slide is generated when you run the `show-disks` command.

Using the `show-nets` Command

Use the `show-nets` command at the `ok` prompt to:

- Display a list of device paths for the Ethernet controllers installed on the system.
- Select the device path that is associated with the controller for which the alias is to be created

```
ok show-nets
a) /pci@7c0/pci@0/pci@2/network@0,1
b) /pci@7c0/pci@0/pci@2/network@0
c) /pci@780/pci@0/pci@1/network@0,1
d) /pci@780/pci@0/pci@1/network@0
q) NO SELECTION
Enter Selection, q to quit: a
/pci@7c0/pci@0/pci@2/network@0,1 has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
      for creating devalias mydev for
      /pci@7c0/pci@0/pci@2/network@0,1
ok
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `show-nets` command displays a list of device paths for the Ethernet controllers installed on the system. You can select the device path that is associated with the controller for which the alias is to be created.

The output presented in the example in the slide is generated when you run the `show-nets` command.

Booting from the diag-device or boot-device Variable

This section covers the following topics:

- Using the diag-device variable
- Using the boot-device variable

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The system can be booted from the device targeted by either the diag-device or boot-device variable, depending on the value of the diag-switch? variable. In this section, you are shown how to use both variables.

Using the diag-device Variable

The diag-device variable:

- Contains the name of the default diagnostic mode boot device
- Has a default value of `net`

Use the `diag-device` variable at the `ok` prompt to set the value of the startup device.

```
ok setenv diag-device disk
diag-device =          disk
ok boot
Boot device: disk  File and args:
Loading ufs-file-system package 1.4 04 Aug 1995 13:02:54.
FCode UFS Reader 1.12 00/07/17 15:48:16.
<output omitted>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `diag-device` variable contains the name of the default diagnostic mode boot device. When the value of the `diag-switch?` variable is set to `true` and the system is powered on, the system boots from the value specified in the `diag-device` variable. The default value of the `diag-device` variable is `net`.

Note: When you type `boot` at the `ok` prompt, the system boots from the default startup device if the `diag-switch?` variable is not set. You can set the value of the startup device in the `boot-device` variable.

In the example in the slide, we set the value of the `diag-device` variable to `disk` and boot the system.

Using the boot-device Variable

The `boot-device` variable:

- Contains the name of the device from which the system boots when the `diag-switch?` variable is set to `false`.
- Contains one or more device specifiers separated by spaces
- Has a default value of `disk net`

Use the `boot-device` variable at the `ok` prompt to specify the boot device.

```
ok setenv diag-switch? false
diag-switch? =          false
ok setenv boot-device disk
boot-device =           disk
ok boot
Rebooting with command: boot
<output omitted>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `boot-device` variable contains the name of the device from which the system boots when the `diag-switch?` variable is set to `false`. The `boot-device` variable contains one or more device specifiers that are separated by spaces. Each device specifier is a device alias.

The boot PROM attempts to open each successive device specifier in the list if the previous device is not available. The system uses the first device specifier that opens successfully. The default value of the `boot-device` variable is `disk net`.

In the example in the slide, you set the value of the `boot-device` variable to `disk` and boot the system.

Quiz

Which command do you use to create an alias name for a device?

- a. `devalias`
- b. `nvalias`
- c. `nvunalias`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

Which command do you use at the `ok` prompt to display the entire device tree?

- a. `show-devs`
- b. `show-disks`
- c. `show-nets`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Before you use the `boot-device` variable to specify the boot device, you must set the `diag-switch?` variable to `true`.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Agenda

- Identifying Oracle server hardware
- Troubleshooting hardware faults
- Service Processor and ILOM
- OpenBoot PROM
- POST
- Solaris-based hardware diagnostics
- Case Study 2

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Using the `prtdiag` Command

The `prtdiag` command displays the following information about a system:

- System configuration, including information about the frequency of the clock, the CPU, memory, and the I/O card types
- Diagnostic information
- Failed field-replaceable units (FRUs)

When the `prtdiag` command is executed, the following exit values are returned:

- **0:** Indicates that no failures or errors are detected in the system
- **1:** Indicates that failures or errors are detected in the system
- **2:** Indicates that an internal `prtdiag` error has occurred in the system

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `prtdiag` command is an OS-level command you run on a system to display diagnostic information about the system. The syntax for the `prtdiag` command is as follows:

```
/usr/platform/`uname -m`/sbin/prtdiag [-v] [-l]
```

The `prtdiag` command supports two options:

- **-v:** Specifies verbose mode. The `-v` option displays the time of the most recent alternate current (AC) power failure and the most recent hardware fatal-error information. The hardware fatal-error information provides detailed diagnostic information about the FRUs. In addition, the `-v` option displays the status of environmental variables, if applicable.
- **-l:** Logs the output. If the `prtdiag` command detects failures or errors on the system, the output is sent to the `syslogd` daemon.

The pages that follow contain examples of `prtdiag` command output. Take a moment to familiarize yourself with each section of the output.

Oracle VTS

- Oracle VTS is an application-level tool that is used for:
 - Hardware testing and validation during development
 - Production validation
 - Receive inspection
 - Troubleshooting
 - Periodic maintenance
- It performs validation tests on:
 - Processors
 - Memory
 - Disk
 - Networks
 - Environment

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle VTS is an Oracle hardware validation test based on the Oracle Solaris operating system. The software's multiple hardware diagnostic tests give administrators the ability to verify the functionality of most hardware controllers and devices for systems based on the SPARC and x86 architectures from the Solaris operating system. The software also provides an infrastructure for developers to develop their own tests and integrate into it.

VTS Testing Modes

- Online stress test mode:
 - Nonintrusive
 - Data-safe
- System exerciser test mode:
 - Intrusive
 - Data-safe
- Component stress test mode:
 - Intrusive
 - Data-safe

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Online Stress Test Mode

Use this mode for testing if you want to simultaneously run other applications in addition to this software on the system. The goal of this mode is to validate the hardware without having significant impact on other applications running on the system. In this mode, the tests would not stress the system, and to provide coverage they might have to run for longer periods of time. If there are any latent faults in the system hardware, the tests at this level try to stimulate those faults before the application is impacted by the fault. When the test stimulates the fault, the OS detects the fault and takes appropriate action. The test characteristics in this mode are as follows:

- **Nonintrusive:** The tests running in this mode do not intrude on the applications that are running on the system. Nonintrusive mode does not change any system configuration that can have an effect on the applications.
- **Data-safe:** The tests running in this mode do not cause any kind of data corruption.
- **Test time:** The time of completion for these tests could vary from milliseconds to days. There is no restriction on the amount of time.
- **Resource usage:** Resource usage is designed so that the test does not have any distinguishable effect on the applications that are running on the system. The goal is to use as few resources as possible for a given instance of time.

System Exerciser Test Mode

This mode provides system exercising capability. It tests the system with all its components. All tests are run in parallel. In this mode, no other application except this software should be running on the system. The characteristics of the tests in this mode are as follows:

- **Intrusive:** The tests can be intrusive in nature. No other application should run on the system while these tests are running.
- **Data-safe:** The tests are data-safe by default.
- **Resource usage:** The tests are aware that there could be other tests running. Accordingly, the tests then allocate and lock resources.

Component Stress Test Mode

This mode performs targeted testing of system components and modules. In this mode, the test puts maximum stress on the component or module under test. The tests are not run simultaneously with other tests; all tests are run sequentially. No other user application should be running on the system. The characteristics of the tests in this mode are as follows:

- **Intrusive:** The tests can be intrusive in nature. The tests might need specific system configuration settings to run efficiently.
- **Data-safe:** The tests are data-safe by default.

Installing Oracle VTS

You obtain Oracle VTS from your IPS publisher.

```
# pkg install SunVTS-incorporation sunvts
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

VTS Troubleshooting Tasks

VTS Task	VTS Commands
Create and save a test session.	<code>./vts_cmd save_session</code> <code>[overwrite:yes/no,defaultMode:test mode,</code> <code>name:session name]</code>
List test sessions.	<code>./vts_cmd list_sessions</code>
Load a test session.	<code>./vts_cmd load_session [defaultMode:testMode,</code> <code>name:sessionName]</code>
Start and stop the test session.	<code>./vts_cmd start</code> <code>./vts_cmd stop</code>
Get VTS test session status.	<code>./vts_cmd get_status</code>
View VTS logs.	<code>cd /var/sunvts/logs</code> <code>more sunvts*</code>
Generate a test report.	<code>vtsreportgenerate</code>



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

VTS Logs

VTS Log	Description
VTS test error log	Contains timestamped VTS test error messages. The log file path name is <code>/var/sunvts/logs/sunvts.err</code> . This file is not created until a VTS test failure occurs.
VTS Verbose log	Verbose messages are the test progression informative messages that are logged to the log file in <code>/var/sunvts/logs/sunvts.verbose</code> . These messages are logged only when the verbose mode is enabled and helps you continuously monitor what is happening during the testing.
VTS Event log	Contains test start messages. Every test that is started by VTS infrastructure logs its start message in the <code>sunvts.event</code> file, which has the start time and version of the test binary. The path name of this log file is <code>/var/sunvts/logs/sunvts.event</code> .
VTS Report log	A report of the VTS test session logged using the <code>vtreportgenerate</code> utility. The log file path name is <code>/var/sunvts/logs/sunvts.rpt</code> .



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

VTS logs are located in the `/var/sunvts/logs` directory.

VTS Messages

Message Type	Log file	Description
INFO	Info log	Displayed when a non-error test event occurs
ERROR	Error and info log	Displayed when a test detects an error, often an indication of a problem with a specific function or feature of the device under test
FATAL	Error and info log	Displayed when a test encounters a severe condition that might cause the test to exit, such as a failure to open a device. These messages generally do not indicate a hardware failure.
VERBOSE	Verbose log	Test progression messages that are displayed when the verbose feature is enabled
WARNING	Info log	These messages warn you of a condition that might prevent the physical test from running with its full capabilities. These warning messages do not necessarily indicate an error.
ALERT		Displayed when a test encounters a software error or a resource issue that causes the test to exit

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are various messages for each of the tests that run under VTS. Most messages contain all the explanatory text that is possible for each event that is encountered. Some messages are information messages (INFO, VERBOSE, WARNING), which do not represent an error, while some messages are error messages (ERROR and FATAL), which indicate that the test detected a failure or was unable to run.

Whenever possible, the error message supplies a message type, followed by the message text, one or more probable causes, and one or more recommended corrective actions. The following table describes the types of messages you might encounter. All messages are displayed in the Message panel, and most are logged in one of the logs (Info or Error).

Agenda

- Identifying Oracle server hardware
- Troubleshooting hardware faults
- Service Processor and ILOM
- OpenBoot PROM
- POST
- Solaris-based hardware diagnostics
- **Case Study 2**

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 2: Hardware Failure

- Clarify problem statement:
 - System `host1` has experienced a system panic!
 - No resent updates or changes to software
- Using “assumptive” method, prioritize the most probable causes of failure:
 - Faulty hardware
- Starting with the most probable cause of failure, gather pertinent information:
 - Peruse system message log.
 - Run system diagnostics.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 2: Troubleshooting Steps

The follows steps are used to identify the cause of the problem:

1. Search `/var/adm/messages` file for error messages.
2. Query the fault manager for faulty components.
3. List the probable cause(s) of the problem.
4. Log in to the service processor.
5. Configure and run POST diagnostics
6. Evaluate POST results to identify failing hardware component.
7. Apply ASR as a temporary solution to the problem.
8. Repair the fault.
9. Verify the solution.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the high-level steps involved in this case study.

Case Study 2: Walkthrough

```
##### Peruse /var/adm/messages file for errors. #####
# more /var/adm/messages
...
Corrected system bus (CE) Event detected by CPU20 at TL=0, errID
    0x002af3bd.3cc195b8
Feb 13 17:56:55 itellin2 AFSR 0x00000002<CE>.00000103 AFAR 0x000000d0.ed685070
Feb 13 17:56:55 itellin2 Fault_PC 0x100260f8 E synd 0x0103 Slot D: J7901
Feb 13 17:56:55 itellin2 SUNW,UltraSPARC-IV: [ID 630551 kern.info] [AFT1]
    kern.warning] WARNING: [AFT1] Uncorrectable Memory Error on CPU20 at TL=0,
    errID 0x002af3bd.3cc195b8.
...
##### A DIMM has failed (Uncorrectable Memory Error). #####
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide begins the Case Study 2 walkthrough. Note that lines beginning with “#####” are comments.

Case Study 2: Walkthrough

```
##### Query fault manager for faulty components. #####
```

```
# fmadm faulty
```

Time	UUID	MSG_ID	Severity
2011-09-21/13:59:35	f13524d6-9970-4002-c2e6-de5d750f4088	ILOM-8000-2V	Major

```
Fault class : fault.memory.dimm
```

```
Fault class : fault.memory.dimm 95%
```

```
Affects : mem:///unum=MB/CMP0/BR0/CH1/D0
```

```
faulted but still in service
```

```
FRU : FRU : mem:///unum=MB/CMP0/BR0/CH1/D0 95%
```

```
faulty
```

Description : The number of errors associated with this memory module has exceeded acceptable levels. Refer to <http://sun.com/msg/SUN4U-8000-2S> for more information.

Response : Pages of memory associated with this memory module are being removed from service as errors are reported.

Impact : Total system memory capacity will be reduced as pages are retired.

Action : Schedule a repair procedure to replace the DIMM. Use Use `fmdump -v -u <EVENT_ID>` to identify the module.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 2: Walkthrough

```
##### Log in to the Service Processor. #####
# ssh root@<SP_IPAddress>
Password: xxxxxx

##### Run firmware-based diagnostics (POST). #####
-> set /SYS/keyswitch_state=diag
-> stop /SYS
Are you sure you want to stop /SYS (y/n)? y
Stopping /SYS
-> start /SYS
Are you sure you want to start /SYS (y/n)? y
Starting /SYS

##### After POST completes, check for faults. #####
-> show faulty
```

Target	Property	Value
/SP/faultmgmt/0	fru	/SYS/MB/CMP0/BR1/CH0/D0
/SP/faultmgmt/0	timestamp	Dec 14 23:01:32
/SP/faultmgmt/0/ faults/0	timestamp	Dec 14 23:01:32 faults/0
/SP/faultmgmt/0/ faults/0	sp_detected_fault	/SYS/MB/CMP0/BR1/CH0/D0 Forced fail (POST)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 2: Walkthrough

```
##### Enable ASR as a temporary solution to the problem. #####
-> set /HOST/diag mode=normal
-> set /HOST/diag level=max
-> set /HOST/diag trigger=power-on-reset
-> start /SP/console
ok setenv auto-boot true
ok setenv auto-boot-on-error? true
ok reset-all

##### Scheduled maintenance replaces faulty DIMM (/SYS/MB/CMP0/BR1/CH0/D0).#####
##### Verify new DIMM using ILOM. #####
-> set /SYS/MB/CMP0/BR0/CH0/D0 component_state=Enabled
-> set /SYS/keyswitch_state=diag
-> stop /SYS
Are you sure you want to stop /SYS (y/n)? y
Stopping /SYS
-> start /SYS
Are you sure you want to start /SYS (y/n)? y
Starting /SYS

##### After POST completes, check for faults. #####
-> show faulty
->

##### Disable ASR. #####
-> start /SP/console
ok setenv auto-boot-on-error? False
ok reset-all
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 3 Overview

- 3-1: Using OBP commands to display system information
- 3-2: Modifying OBP variables

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- List components commonly found in Oracle enterprise-class systems
- Identify faulty hardware behavior
- Identify the best method for troubleshooting hardware faults
- Determine the best tools for analyzing and troubleshooting hardware faults
- Describe how to use the Server Processor (SP) when troubleshooting hardware faults
- Determine how to use the OpenBoot PROM when troubleshooting SPARC-related hardware faults
- Describe Solaris-based hardware diagnostics

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

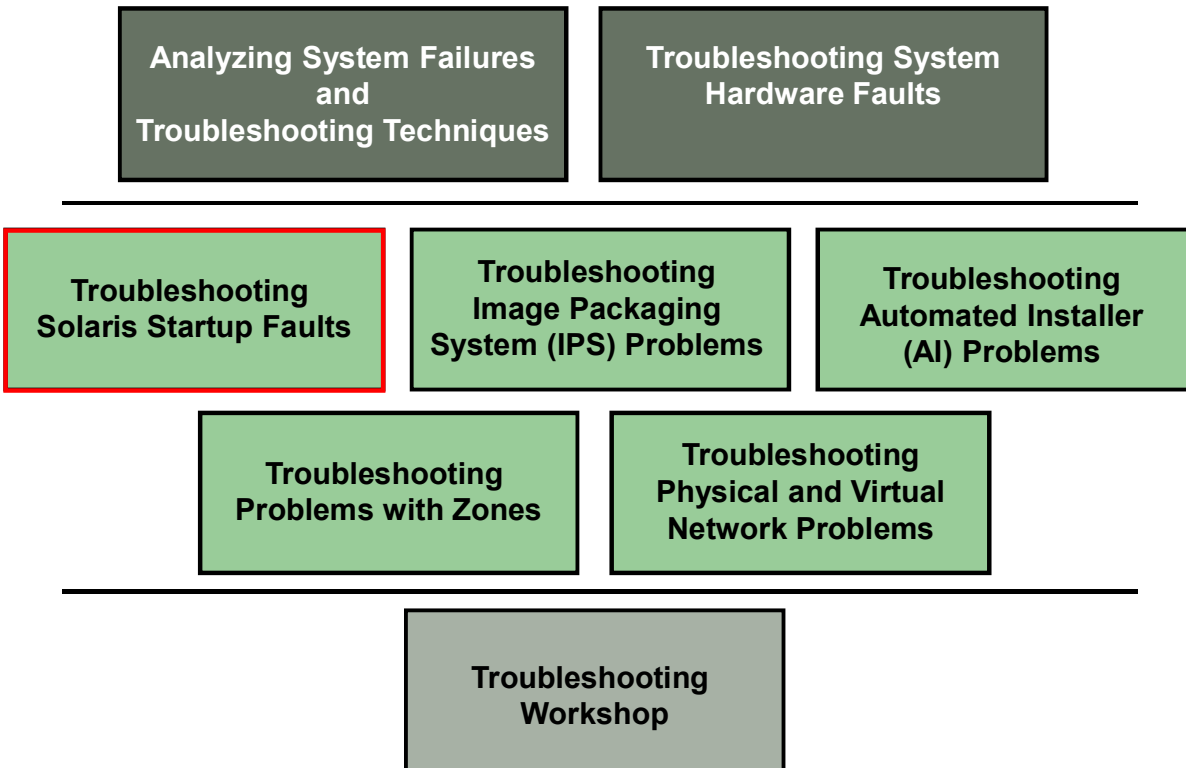
4

Troubleshooting Solaris Startup Faults

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this lesson, “Troubleshooting Solaris Startup Files”, you are introduced to the SPARC Open Boot Prom (OBP) and POST features. You are also presented with potential SPARC-based system booting issues, their potential causes, and shown how to troubleshoot and resolve these issues.

Objectives

After completing this lesson, you should be able to:

- Identify the stages associated with a successful Solaris OS startup
- List faults common to Solaris OS startup failures
- Identify the best method for troubleshooting Solaris OS startup failure
- Determine best tools for analyzing and troubleshooting Solaris OS startup failures
- Identify the best tools and method for analyzing and troubleshooting failures associated with system services
- Determine when and how to use Solaris Boot Environments (BEs) for quick operating system recovery

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

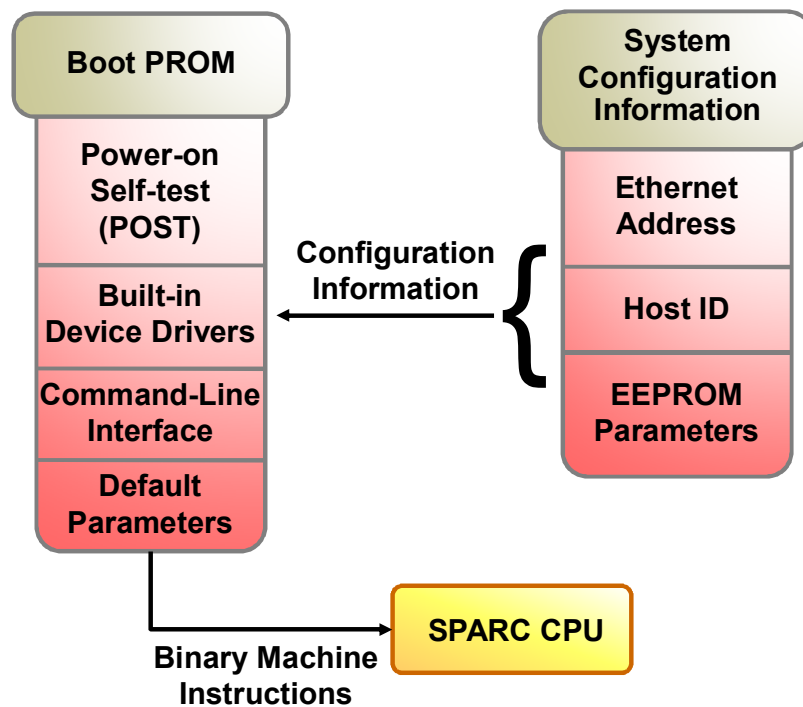
Agenda

- Oracle Solaris 11 startup
- Troubleshooting Solaris startup faults
- Service Management Facility (SMF)
- Troubleshooting SMF faults
- Solaris Boot Environment (BE)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Boot PROM Initialization Sequence

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

Note: When the boot process has completed and the operating system is running, you see a login prompt displayed on the console. When the operating system is not running, the `ok` prompt is displayed.

Boot Process

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In general, when a system is powered on, the PROM monitor on SPARC systems and the BIOS on x86/x64 systems run a POST procedure that checks the hardware and memory on the system. If no errors are found, the system automatically begins the boot process.

The boot process consists of six independent phases:

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

- **Kernel phase:** During this phase, Oracle Solaris is initialized and a minimal root file system is mounted on the ramdisk that was constructed from the boot archive. In some environments, such as an installation, the ramdisk is used as the root (/) file system and remains mounted. The ramdisk contains a set of kernel files and drivers that is sufficient to mount the root file system on the specified root device. The kernel then extracts the remainder of the primary modules from the boot archive, initializes itself, mounts the real root file system, and then discards the boot archive. In addition, the kernel runs the `/sbin/init` program, which in turn starts the `init` daemon. By default on SPARC platforms, the kernel is in `/platform/`uname -m`/kernel/unix`. The default kernel on x86 platforms is `/platform/i86pc/kernel/amd64/unix`.
Note: If a boot process fails because of an unusable `/etc/system` file, on a SPARC system, issue the interactive boot command: `boot -a`. When the system prompts you to enter the name of the system file, type in the name of your backup `system` file, or, alternatively, enter `/dev/null` for a null configuration file.
- **init phase:** During this phase, the `init` daemon initializes stream modules, sets up the system for a correct response to a power-fail shutdown, and starts the `svc.startd` daemon.
- **svc.startd phase:** During this phase, the `svc.startd` daemon starts the system processes.

How Oracle Solaris Boot Archives Are Managed

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

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A boot archive is a subset of a root file system and contains all of the kernel modules, `driver.conf` files, in addition to a few configuration files. These files are located in the `/etc` directory. The 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 files in the SPARC boot archive are located in the `/platform/`uname -m`/archive_cache` directory. To list the contents of the boot archive, use the `bootadm list-archive` command.

Note: The files that are part of the x86 boot archive are located in the `/platform/i86pc/amd64/archive_cache` directory and can be viewed using the `bootadm list-archive` command.

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

Note

In the Oracle Solaris 11.1 release, the `bootadm` commands has been modified on x86-based systems, and some new subcommands have been added. to enable you to perform most of the administrative tasks that were previously done by editing the `menu.lst` file. These tasks include managing the GRUB menu, setting kernel arguments for a specific boot entry, and managing the boot loader.

SMF manages the following boot archive services:

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

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

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

Agenda

- Oracle Solaris 11 startup
- Troubleshooting Solaris startup faults
- Service Management Facility (SMF)
- Troubleshooting SMF faults
- Solaris Boot Environment (BE)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Booting in Verbose Mode

Use the OBP command `boot -v` to boot in verbose mode.

```
ok> boot -v

SC Alert: Host System has Reset

Sun Fire T200, No Keyboard
Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #67263734.
Ethernet address 0:14:4f:2:5c:f6, Host ID: 84025cf6.

Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a File and args: -v
module /platform/sun4v/kernel/sparcv9/unix: text at [0x1000000, 0x10d3297] data at 0x10000000
module /platform/sun4v/kernel/sparcv9/genunix: text at [0x10d3298, 0x1334b9f] data at 0x1063ed40
module /platform/SUNW,Sun-Fire-T200/kernel/misc/sparcv9/platmod: text at [0x1334ba0, 0x1334d9f] data
at 0x1069c090
module /platform/sun4v/kernel/cpu/sparcv9/SUNW,UltraSPARC-T1: text at [0x1334da0, 0x133798f] data at
0x1069c740
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Ethernet address = 0:14:4f:2:5c:f6
mem = 16646144K (0x3f8000000)
avail mem = 16462487552
root nexus = Sun Fire T200
...
cpu0: UltraSPARC-T1 (chipid 0, clock 1000 MHz)
cpu1: UltraSPARC-T1 (chipid 0, clock 1000 MHz)
cpu2: UltraSPARC-T1 (chipid 0, clock 1000 MHz)
cpu3: UltraSPARC-T1 (chipid 0, clock 1000 MHz)
cpu4: UltraSPARC-T1 (chipid 0, clock 1000 MHz)
...
console login:
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you are experiencing Solaris startup problems, a good place to start is boot in verbose mode. On SPARC-based machines, you use the OBP command `boot -v`. This also provides you with good debug information.

Booting from an Incorrect Device or Disk

This section covers the following topics:

- Booting from an incorrect device type
- Booting from an incorrect disk

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Booting from an Incorrect Device Type

Possible cause #1: The `diag-switch?` parameter is set to `true`. To set the `diag-switch?` parameter to `false`:

- Type the following command at the `ok` prompt:
`ok setenv diag-switch? false`
`ok boot`

Possible cause #2: The `boot-device` parameter is set to `net` instead of `disk`. To set the `boot-device` parameter to `disk`:

- Type the following command at the `ok` prompt:
`ok setenv boot-device disk`
`ok boot`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Consider a scenario in which a SPARC-based system that should boot from the disk boots from the network. Two possible causes of the problem are presented in the slide. To have the system boot from the disk, complete one of the set of steps presented in the slide.

Note: The commands presented in the slide cause the system to boot from the disk defined as `disk` in the list of device aliases.

Booting from an Incorrect Disk

Possible cause #1: The `boot-device` parameter is not set to the correct disk. To set the `boot-device` parameter to the correct disk:

Type the following command at the `ok` prompt:

```
ok setenv boot-device disk
ok boot
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Consider a scenario in which a SPARC-based system boots from the wrong disk. For example, there is more than one disk in the system. The system needs to boot from the disk `disk2`. However, the system boots from the disk `disk1`. To set the `boot-device` parameter to `disk2`, follow the steps presented in the slide.

Recovering a Corrupted Boot Block

1. Install a boot block for a ZFS root file system by using the `bootadm install-bootloader` command.
2. Verify that the boot blocks are installed by rebooting the system.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If a boot image becomes corrupted, you can recover the Solaris image by performing the steps presented in the slide. These steps are used to install boot blocks for a ZFS root file system on both SPARC and x86 based systems.

Note: If you physically replace the disk that is intended for the root pool and the Oracle Solaris OS is then reinstalled, or you attach a new disk for the root pool, the boot blocks are installed automatically. If you replace a disk that is intended for the root pool by using the `zpool replace` command, then you must install the boot blocks manually so that the system can boot from the replacement disk.

Booting a System with the Kernel Debugger (kmdb) Enabled

1. Halt the system cleanly by using the `halt` command.
2. Type `boot -k` to request the loading of the kernel debugger and press return.
3. Access the kernel debugger as follows:
If you are using a serial console, send a break by using the method that is appropriate for your type of serial console.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Running a system under the kernel debugger can be very helpful if you need to troubleshoot system problems. The kernel debugger can help you investigate system hangs. For example, if you are running the kernel while the kernel debugger is active, and you experience a hang, you might be able to break into the debugger to examine the system state. Also, if the system panics, the panic can be examined before the system is rebooted. In this way, you can get an idea of which section of code might be causing the problem.

The steps presented in the slide show how to load the kernel debugger (`kmdb`) on a SPARC-based system.

Note: Use the `reboot` command and the `halt` command with the `-d` option if you do not have time to debug the system interactively. Running the `halt` command with the `-d` option requires a manual reboot of the system afterward. However, if you use the `reboot` command, the system boots automatically.

Note for step 1: This action displays the `ok` prompt.

Note for step 3: A welcome message is displayed when you enter the kernel debugger for the first time.

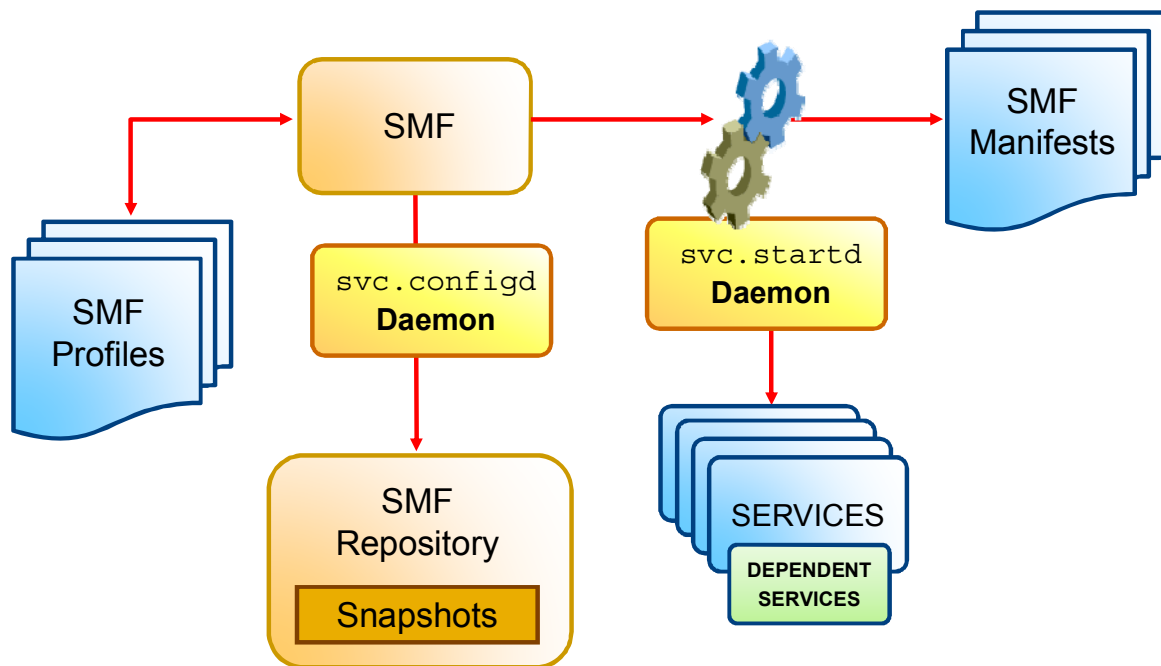
Agenda

- Oracle Solaris 11 startup
- Troubleshooting Solaris startup faults
- **Service Management Facility (SMF)**
- Troubleshooting SMF faults
- Solaris Boot Environment (BE)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

SMF Architecture and Features Review



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When a system is booted, the SMF consults the SMF profiles to determine which services should be enabled. The SMF then starts the `svc.startd` daemon, which in turn consults the SMF manifests to gather property and instance information about each service before starting each service and its associated dependents. The SMF uses the Service Configuration Repository (also known as the SMF Repository) to store state and configuration information about each service instance in addition to per-service snapshots that are taken at the time each service is successfully started and used as backups. The SMF repository is managed by the `svc.configd` daemon.

Monitoring SMF Services

Task	Command
List all the services currently running on the system.	<code>svcs</code>
List explanations for service states and in verbose mode	<code>svcs -xv</code>
Display the status of a service instance.	<code>svcs -l FMRI</code>
Display the service dependents.	<code>svcs -D FMRI</code>
Display the dependencies of a service.	<code>svcs -d FMRI</code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You monitor services with the `svcs` command and a variety of useful troubleshooting options, as presented in the slide.

Notes for listing service information: The `svcs` command displays information about service instances as recorded in the service configuration repository.

Notes for displaying the status of a service instance: 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.

Notes for displaying the service dependents: Sometimes, services depend on one another to operate properly. If a service has dependents and the service fails, the services that depend on that service are affected. Whenever an issue with a service occurs that requires administrative intervention (such as taking the service down for maintenance), one of the first actions that is taken is to see what dependents that service has. To determine which service instances depend on another service, use the `svcs -D` command.

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

Notes for displaying the dependencies of a service: To determine the services on which a service instance depends, use the `svcs -d` command.

Determining System Readiness Using Milestones

- A milestone is a new mechanism made to be compatible with run-levels found in earlier releases of Solaris.
- Milestones define a specific state of system readiness.
- System services can use milestones as dependencies to ensure that the system is ready before the service attempts to come online.
- The system can only be in one of the following milestones at any time: none, single-user, multi-user, multi-user-server, or all.
- Milestone manifest files are located in `/lib/svc/manifest/milestone`.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A milestone is a new mechanism made to be compatible with run-levels found in earlier releases of Solaris. Milestones define a specific state of system readiness, each with a specific set of services enabled. System services can use milestones as dependencies to ensure that the system is ready before the service attempts to come online. Before SMF, system boot progresses from run level S to 3, executing scripts in `/etc/rc#.d` along the way. The system can only be in one of the following milestones at any time: none, single-user, multi-user, multi-user-server, or all. The system's current milestone is defined in the `options_ovr/milestone` property of the `svc:/system/svc/restarter:default` service. Note that Solaris networking has its own milestone called `network`.

Identifying Service States

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

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

Setting Up Service State Transition Notifications

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

To set up the notifications:

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

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

Installing the smtp-notify Package

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

```
# pkg install service/fault-management/smtp-notify
      Packages to install:      1
    Create boot environment:    No
      Services to restart:      1

DOWNLOAD                                PKGS      FILES      XFER (MB)
Completed                             1/1        4/4        0.0/0.0

PHASE                                ACTIONS
Install Phase                         24/24

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

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

Enabling the `smtp-notify:default` Service

To enable the SMF notification service:

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

Confirm the service is online and running:

```
# svcs | grep smtp
online      10:33:31 svc:/system/fm/smtp-notify:default
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After you have installed the `smtp-notify` package, enable the service and then confirm it is online.

Note: `smtp-notify` is dependent upon `sendmail` and/or `sendmail-client` services (shown by `svcs -xv smtp-notify`) so they may need to be enabled first, if they are offline.

Configuring Service State Transition Notifications

To configure the service state transition notifications for all services:

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

To configure notifications for a single service:

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

Note

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

To configure notifications for a single service, you use the same command but you specify the FMRI for a specific service; you don't need to specify the `-g` option before identifying the service state change that you want to monitor. In the second example, you are setting up a notification to alert you if the `apache22` service state changes from online to any other state.

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

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

Service State Transition Notification Example

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

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

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

Service State Transition Notification Example

```
<continued from previous slide>

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

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

? q
#
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

Managing Service State Transition Notifications

To view configured notifications:

```
# 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:

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

Managing SMF Services

Task	Command
Disable a service.	<code>svcadm disable <i>FMRI</i></code>
Enable a service.	<code>svcadm enable <i>FMRI</i></code>
Clear current fault.	<code>svcadm clear <i>FMRI</i></code>
Refresh a service.	<code>svcadm refresh <i>FMRI</i></code>
Restart a service.	<code>svcadm restart <i>FMRI</i></code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You manage services with the `svcadm` command. The subcommands for stopping, starting, refreshing, and restarting a service are presented in the slide.

Notes for disabling a service: 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. When you have disabled the service, the service status change is recorded in the service configuration repository. The disabled state will persist across reboots. This means that the only way to get the service running again is to enable it. You can also temporarily disable a service with the `-t` option. With this option, the service returns to an online state on reboot.

Notes for enabling a service: As with the `svcadm disable` command, the `svcadm enable` command, 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.

Notes for refreshing a service: When a refresh is done, 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.

SMF Configuration Repository

- Stores state and configuration information about each service instance
- Is located in `/etc/svc/repository.db`
- Is managed by the `svc.configd` daemon
- Provides a consistent and persistent way to enable or disable a service
- Provides a consistent view of service state

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

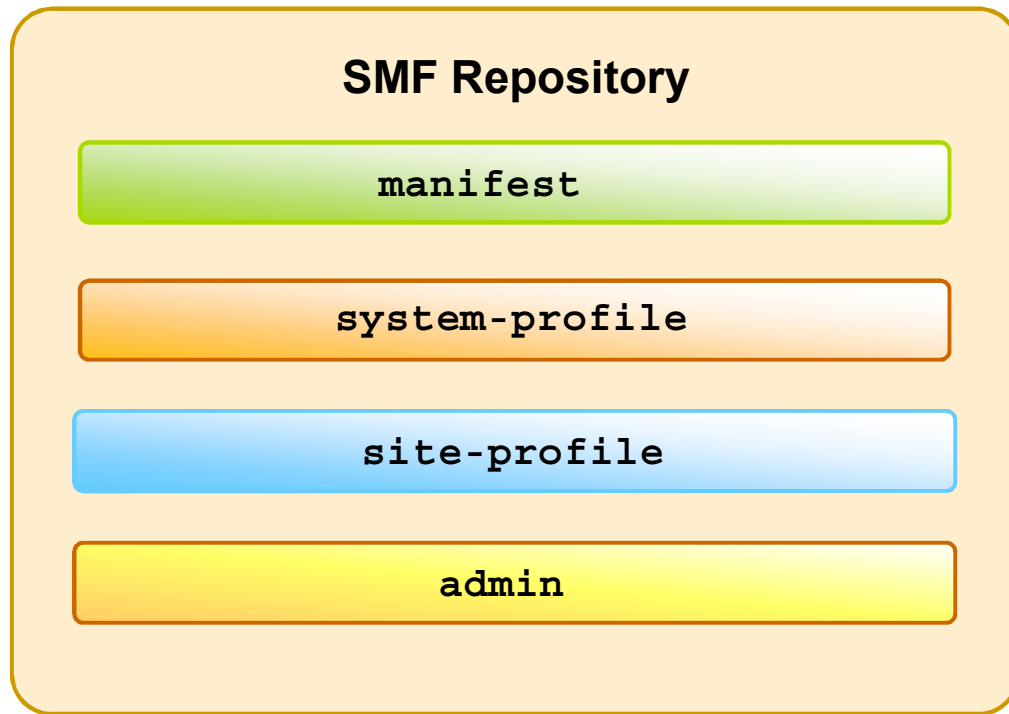
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

In turn, the repository provides a consistent and persistent way to enable or disable a service, as well as a consistent view of the service state. This capability helps you to debug service configuration problems.

Exploring the SMF Layers



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The SMF repository consists of four layers that can be used to help determine which settings have been customized by an administrator and which settings are delivered by the software. The four layers are as follows:

- **manifest:** Imported full manifest files that completely define a service or an instance, that is located in a standard location: `/lib/svc/manifest` or `/var/svc/manifest`
- **system-profile:** Specifically named profiles (`/etc/svc/profile/generic.xml` or `/etc/svc/profile/platform.xml`) that are applied to the system and delivered by the Solaris consolidations
- **site-profile:** Profiles that are site specific and are either applied from the `/etc/svc/profile/site` directory or from the `/etc/svc/profile/site.xml` or `/var/svc/profile/site.xml` file
- **admin:** Administrative customizations to the system done with `svccfg add/set/del` subcommands as well as through enabling/disabling services through the command line. Manifests and profiles imported and applied from nonstandard locations (that is, outside of `/lib/svc/manifest` or `/var/svc/manifest`) are considered customizations and are brought in at the `admin` layer.

The layers are hierarchical, with the `admin` layer taking precedence. If a property has a value in the `admin` layer, that value will be used by the service. If not, the `site-profile` layer is consulted, and then the `system-profile` layer, and eventually the `manifest` layer. This behavior allows for local changes to take precedence over the default settings.

The system automatically manages these layers. Any direct changes that you as the system administrator make to the repository appear only in the `admin` layer. Other layers are changed only by placing or removing files in standard locations. When a property is put into the repository because of file contents, the information about that property includes the name of that file.

Note: You can use the `svccfg listprop` command to explore layers. You can use the `svccfg listcust` command only to list customizations.

Service Bundles: Manifests and Profiles

There are two types of service bundles: manifests and profiles.

- Manifest files location: `/lib/svc/manifest`
- Manifests describe services and instances of a service, including:
 - Property groups
 - Properties
 - Dependencies
 - Methods
- Profile files location: `/etc/svc/profiles`
- Profiles provide customization of a service or an instance of a service, including:
 - Should service be enabled or disabled
 - Modifications to service configuration properties

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

At the core of SMF is the concept of a *service bundle*, an XML-based file that is used to describe everything about a service or an instance of a service—including its configuration—prior to it being imported into the SMF configuration repository. Service bundles are used to deliver services into Oracle Solaris, but they also act as a useful way for administrators to deliver custom configuration across a variety of systems.

Two types of service bundles are used in SMF: manifests and profiles. Both are based on an XML DTD found in `/usr/share/lib/xml/dtd/service_bundle.dtd.1`.

Viewing SMF Repository Information

- **svccprop:** Lists the values assigned to property groups or properties in running snapshot
- **svccfg listpg:** Lists information about property groups in the selected service, service instance, or snapshot
- **svccfg listprop:** Lists information about properties and property group in the selected service, service instance, or snapshot
- **svccfg listcust:** Lists any site customizations in the site-profile or admin layer

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use the **svccfg** and **svccprop** commands to view information in the service configuration repository.

- **svccprop:** Because this command combines data for the service, and service instance, it provides a comprehensive or “composed” view of the data.
- **svccfg listpg:** All property group names, types, and flags are listed.
- **svccfg listprop:** For property groups, the names, types, and flags are listed. For properties, the names, types, and values are listed.
 - **-l layer_name:** Lists the properties and property groups within the named layer in a service or service instance. Using all as a layer name lists all of the layers and properties for that service. Note that each service and service instance is displayed separately.
 - **-f:** Lists the file name that a property came from.
 - **-o:** Selects the fields to display.
- **svccfg listcust:** Also listed are any masked entries for the selected service or service instance.
 - **-M:** Lists only masked entities.
 - **-L:** Shows all local customizations, which includes both administrative customizations and site profile customizations

Quiz

Which command can you run to quickly ascertain what services are not running due to errors?

- a. `svcs -a`
- b. `svcadm enable`
- c. `svcadm refresh`
- d. `svcs -xv`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Agenda

- Oracle Solaris 11 startup
- Troubleshooting Solaris startup faults
- Service Management Facility (SMF)
- **Troubleshooting SMF faults**
- Solaris Boot Environment (BE)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Debugging a Service That Is Not Starting

1. Request information about the failed service by using `svcs -xv`.
2. Verify the failure by disabling and enabling the service by using
`svcadm disable serviceinstance`
`svcadm enable serviceinstance`
3. Read the log associated with the failing service to identify the cause of the failure.

```
# svcs -xv
svc:/application/pkg/server:default (image packaging
  repository)
  State: maintenance since Mon 15 Apr 2013 02:20:37 PM PDT
  Reason: Start method failed repeatedly, exit with status 1.
  See: http://support.oracle.com/msg/SMF-8000-KS
  See: /var/svc/log/application-pkg-server:default.log
Impact: This service is not running.
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you have a service that is disabled and not starting, you can debug it by using the steps shown in the slide.

Note for step 1: The `-x` option provides additional information about the service instances that are affected.

In the example, the IPS service has failed to start. To find out more about the problem, you run the `svcs -xv` command for the service. The output for the `svcs -xv` command provides the following information:

- **State:** The state of the service and the date and time stamp
- **Reason:** The service has been placed in “maintenance” state due to repeated startup failures.
- **See:** The URL to a knowledge article on the issue
- **See:** The log containing information about the failure
- **Impact:** What services have been affected by the problem

Interpreting SMF Log Files and Trees

- Service-specific information are logged in individual files for each service instance in `/var/svc/log`.
- The `svcs -x service` command is used to determine the name of a service's log file.
- SMF writes log messages only if administrative intervention is required.
- Check to see what services depend on the service in question:
`svcs -D FMRI`
- Check to see what dependencies the service has:
`svcs -d FMRI`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Service-specific information, including errors the service or its methods emits, as well as information about enable actions, start times, and so on, are logged in individual files for each service instance in `/var/svc/log`. To determine the name of a service's log file, run the `svcs -x service` command.

By default, SMF writes log messages to the `syslog` program and the console only if administrative intervention is required, for example, if a service enters the maintenance state.

Interpreting SMF Log Files and Trees Example

```
# cd /var/svc/log
# more application-pkg-server:default.log
...
ppriv -s A=basic,-file_link_any,-proc_info,-
      proc_session,net_privaddr -e /usr/lib/pkg.depotd --cfg
      svc:/application/pkg/server:default
pkg.depotd: unable to bind to the specified port: 80. Reason:
      Port 80 is in use on '127.0.0.1'; perhaps the previous
      httpserver did not shut down properly.
[ Mar 27 11:04:27 Method "start" exited with status 1. ]
[ Mar 27 11:04:27 Executing start method
      ("//lib/svc/method/svc-pkg-depot start"). ]
Dropping net_privaddr privilege
. . . . .
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right of the "E".

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here you see the reason for the IPS service failing to start.

Troubleshooting Service Dependencies

- Dependencies for a service are defined in the service manifest.

```
...
    <dependency
        name='loopback'
        grouping='require_all'
        restart_on='none'
        type='service'>
        <service_fmri value='svc:/network/loopback' />
    </dependency>
    <dependency
        name='physical'
        grouping='require_all'
        restart_on='none'
        type='service'>
        <service_fmri
value='svc:/network/physical:default' />
        </dependency>
....
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

One of the most common causes of service startup failure is due to its dependencies. Services often depend on other services to function properly or services may not function properly in the presence of other services. You can determine a service's dependencies by finding the `dependency` elements in the service's manifest file located in the `/lib/svc/manifest` directory. Make sure that the failing service's dependencies are functioning in accordance with the `grouping` property. Some of the values for the `grouping` property are as follow:

- The `require_all` property – Specifies that all services marked with this grouping must be online before this service came online
- The `require_any` property – Specifies that any of the services in this grouping suffice. This service will go online if one of them is online.
- The `optional_all` property – Specifies the services that are optional for this service. The service works with or without them.
- The `exclude_all` property – Specifies the services which may have conflict with this service. The service will not go online if any exist.

The example shown here is taken from the `network` milestone service manifest file. In this example, in order for the system network to be in a state of readiness, the `loopback` and `physical` services must be online (`grouping='required_all'`). Once the milestone is successfully met other network services, such as: DNS, apache, and ssh, can use the `network` mile service as their dependency. Note that you can determine a services dependencies by using the `svcs -d` (list of services on which this services depends) and `svcs -D` (list of services that are dependent on this service) commands.

Troubleshooting Service Methods

- Service methods determine how a service is started, refreshed, and stopped.

```
...  
<exec_method  
  type='method'  
  name='start'  
  exec='/opt/site/first-boot-script.sh'  
  timeout_seconds='360'  
  <method_context>  
    <method_credential user='root' />  
  </method_context>  
</exec_method>  
  
<exec_method  
  type='method'  
  name='stop'  
  exec=':true'  
  timeout_seconds='60'  
...  

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Service methods determine how a service changes state such as starting, refreshing, and stopping. Changing a service's state often involves executing a program (such as script) and who may run the program (credentials). A service startup failure might related to the executable defined the "start" method.

Using SMF Repository Backups

- SMF automatically takes these backups:
 - Boot backup: Taken immediately before the first change to the repository is made during each system startup
 - `manifest_import` backups: Occur after `svc:/system/early-manifest-import:default` or `svc:/system/manifest-import:default` completes
- System maintains four copies of each type.
- Backups are stored as `/etc/svc/repository-type-YYYYMMDD_HHMMSS` for the date and time when the backup was taken.
- Repository can be restored from these backups.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The SMF automatically takes the following backups of the repository:

- The boot backup is taken immediately before the first change to the repository is made during each system startup.
- The `manifest_import` backups occur after `svc:/system/early-manifest-import:default` or `svc:/system/manifest-import:default` completes, if the service imported any new manifests or ran any upgrade scripts.

Four backups of each type are maintained by the system. The system deletes the oldest backup when necessary. The backups are stored as `/etc/svc/repository-type-YYYYMMDD_HHMMSS`, where `YYYYMMDD` (year, month, day) and `HHMMSS` (hour, minute, second) are the date and time when the backup was taken. Note that the hour format is based on a 24-hour clock.

You can restore the repository from these backups, if an error occurs.

SMF Repository Integrity Check Process

Message is sent to console if integrity check fails:

```
<MESSAGE DISPLAYED BY SMF>
svc.configd: smf(5) database integrity check of:

    /etc/svc/repository.db

failed.  The database might be damaged or a media error might have
prevented it from being verified.  Additional information useful to
your service provider is in:

    /etc/svc/volatile/db_errors

The system will not be able to boot until you have restored a working
database.  svc.startd(1M) will provide a sulogin(1M) prompt for
recovery purposes.  The command:

    /lib/svc/bin/restore_repository

can be run to restore a backup version of your repository.  See
http://sun.com/msg/SMF-8000-MY for more information.
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When the repository daemon, `svc.configd`, is started, it does an integrity check of the configuration repository. If the integrity check fails, the `svc.configd` daemon writes a message to the console similar to the one shown in the slide. The `svc.startd` daemon then exits and starts `sulogin` to enable you to perform maintenance as shown on the next page.

Note: The repository can become corrupted due to one of the following reasons:

- Disk failure
- Hardware bug
- Software bug
- Accidental overwrite of the file

Repairing a Corrupt Repository Example

```
# /lib/svc/bin/restore_repository
. . . . .
The backups are named based on their type and the time what they were taken.
Backups beginning with "boot" are made before the first change is made to
the repository after system boot. Backups beginning with "manifest_import"
are made after svc:/system/manifest-import:default finishes its processing.
The time of backup is given in YYYYMMDD_HHMMSS format.
Please enter either a specific backup repository from the above list to
restore it, or one of the following choices:

      CHOICE              ACTION
      -----
      boot                restore the most recent post-boot backup
      manifest_import     restore the most recent manifest_import backup
      -seed-              restore the initial starting repository (All
                          customizations will be lost, including those
                          made by the install/upgrade process.)
      -quit-              cancel script and quit
Enter response [boot]: boot
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Running this script takes you through the necessary steps to restore a non-corrupt backup. SMF automatically takes backups of the repository at key system moments. The information on the screen guides you through the process. Select boot to restore the most recent post-boot backup.

See <http://support.oracle.com/msg/SMF-8000-MY> for more information on the use of this script to restore backup copies of the smf(5) repository.

The omitted output from above:

If there are any problems which need human intervention, this script will give instructions and then exit back to your shell.

```
/lib/svc/bin/restore_repository[71]: [: /: arithmetic syntax error
```

The following backups of /etc/svc/repository.db exist, from oldest to newest:

```
boot-20000227_221002
manifest_import-20000227_221506
manifest_import-20000227_221703
boot-20000228_095148
boot-20000228_095801
```

Repairing a Corrupt Repository Example

```
After confirmation, the following steps will be taken:

svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
-- renamed --> /etc/svc/repository.db_old_20000228_102111
/etc/svc/repository-boot
-- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).

Proceed [yes/no]? yes

Quiescing svc.startd(1M) and svc.configd(1M): done.
/etc/svc/repository.db
-- renamed --> /etc/svc/repository.db_old_20000228_102111
/etc/svc/repository-boot
-- copied --> /etc/svc/repository.db

The backup repository has been successfully restored.

Rebooting in 5 seconds.
Feb 28 10:30:13 host03 reboot: initiated by root on /dev/console
Feb 28 10:30:19 host03 syslogd: going down on signal 15
syncing file systems... done
rebooting...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Afterwards, the system will reboot and you should confirm you can login and check that the repository is repaired.

Using SMF Repository Snapshots

- Snapshots are taken per service at the time a service is successfully started.
- Standard snapshots include:
 - `initial`: Taken on the first import of the manifest
 - `running`: Used when the service methods are executed
 - `start`: Taken at the last successful start
- SMF service always executes with the `running` snapshot.
- Current property values for a service are incorporated into the `running` snapshot with the `svcadm refresh` command.
- Instance configurations can be viewed or reverted to in a previous snapshot by using the `svccfg` command.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The service configuration repository provides a per-service snapshot at the time each service is successfully started so that fallback is possible. The standard snapshots that are stored in the SMF repository are listed in the slide.

The SMF service always executes with the `running` snapshot. This snapshot is automatically created if it does not exist.

When you change the property values of a service, the changes are incorporated into the `running` snapshot when you execute the `svcadm refresh` command. You can use the `svccfg` command to view or revert to instance configurations in a previous snapshot.

Reverting to an SMF Snapshot

1. Run the `svccfg` command.
 - a. Select the service instance that you want to fix.
 - b. Generate a list of available snapshots by using `listsnap`.
 - c. Select to revert to the `start` snapshot by using `revert start`.
 - d. Quit `svccfg` by using `quit`.
2. Update the information in the service configuration repository by using `svcadm refresh FMRI`.
3. Restart the service instance by using `svcadm restart FMRI`.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If the service's administrative customizations are wrong, you can fix the problem by reverting to the last snapshot that started successfully. The steps for how to revert to a previous SMF snapshot are shown in the slide.

Note for step 1a: You must use an FMRI that fully defines the instance. No shortcuts are allowed.

Note for step 1c: The start snapshot is the last snapshot in which the service successfully started.

Note for step 2: This step updates the repository with the configuration information from the start snapshot.

Note: None of the file-backed properties (that is, properties delivered via manifests or profiles) from the snapshot are restored. Instead, all the administrative customizations in the current configuration are removed, and then all the administrative customizations from the selected snapshot are propagated forward.

Reverting to an SMF Snapshot Example

```
# svccfg
svc:> select system/console-login:default
svc:/system/console-login:default> listsnap
initial
last-import
previous
running
start
svc:/system/console-login:default> revert start
svc:/system/console-login:default> quit
# svcadm refresh system/console-login:default
# svcadm restart system/console-login:default
# svcs console-login:default
online 18:15:32 svc:/system/console-login:default
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example shown in the slide, it is assumed that the `console-login:default` service is in the `maintenance` state. To resolve the issue, you have decided to revert to a previous SMF snapshot to bring the service back online. You have selected the `start` snapshot.

Note: The version of the snapshot you choose to use is based on what you are trying to accomplish.

When you have selected the type of snapshot you want, you quit the service configuration. You then refresh and restart the service. Your final step is to verify that the service is back online.

Observing the SMF Portion of the Boot Process

- To step through the SMF portion of the boot process, boot using the `boot -m milestone=none` command.
- Then step through the milestones for the different boot levels:
 - Milestone `svc:/milestone/single-user:default`
 - Milestone `svc:/milestone/multi-user:default`
 - Milestone `svc:/milestone/multi-user-server:default`

```
ok boot -m milestone=none
console login: root
Password: XXXXXXXX
...
# svcadm milestone svc:/milestone/single-user:default
...
# svcadm milestone svc:/milestone/multi-user:default
...
# svcadm milestone svc:/milestone/multi-user-server:default
...
```

- Use the `svcs` command to check service status and dependencies.
- Use `svccprop -p start` to determine startup properties.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If problems with starting services occur, sometimes a system will hang during the boot. This procedure shows how to troubleshoot this problem.

Several things should be examined if a service fails to start:

- Is the service in maintenance mode? (`svcs -l FMRI`)
- If so, why? Check the log file specified in the `svcs -l FMRI | grep logfile` output, and run `svcs -xv FMRI`.
- If the problem has been resolved, clear the fault with `svcadm clear FMRI`.
- Check for service dependencies with `svcs -d FMRI` The output from `svcs -l` distinguishes between optional and mandatory dependencies.
- Check the startup properties with `svccprop -p start FMRI`.

Adding Truss to a Service

- You can `truss` the startup of a failing service to get some visibility into where it is failing.
- Inserting a `truss` command into the `start/exec` statement for the service:

```
# svccfg -s new_service
svc:/application/new_service> listprop start
start                                method
start/exec                          astring  "/lib/svc/method/new_service start"
start/timeout_seconds               count    60
start/type                          astring  method
svc:/application/new_service> setprop start/exec = "truss -f -a -o
/tmp/truss.out /lib/svc/method/new_service start"
svc:/application/new_service> ^D
# svcadm refresh new_service
# svcadm restart new_service
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `truss` command is a useful tool for analyzing service failures. You can `truss` the startup of a failing service to get some visibility into where it is failing. You do this by inserting a `truss` command into the `start` or `exec` statement for the service. To do this, add `truss` to the beginning of the `start/exec` property with an `svccfg -s` command. For example:

```
# svccfg -s new_service
svc:/application/new_service> listprop start
start                                method
start/exec                          astring  "/lib/svc/method/new_servi
ce start"
start/timeout_seconds               count    60
start/type                          astring  method
svc:/application/new_service> setprop start/exec = "truss
-f -a -o /tmp/truss.out /lib/svc/method/new_service start"
svc:/application/new_service> ^D
# svcadm refresh new_service
# svcadm restart new_service
```


Quiz

Which command can you run to list the properties and property groups within the named layer in a service or service instance?

- a. `svccfg listprop -l layer_name`
- b. `svccfg listcust`
- c. `svccfg listpg`
- d. `svcprop`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Agenda

- Oracle Solaris 11 startup
- Troubleshooting Solaris startup faults
- Service Management Facility (SMF)
- Troubleshooting SMF faults
- Solaris Boot Environment (BE)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Solaris Boot Environments (BEs)

- A BE is a bootable instance of the Oracle Solaris operating system image plus any other application software packages installed into that image.
- BE is an important feature in your system recovery strategy.
- Use an Solaris boot environment as a backup and as a place to test and validate potential fixes.
- You can create BEs from existing ZFS snapshots.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A boot environment is a bootable instance of the Oracle Solaris operating system image plus any other application software packages installed into that image. System administrators can maintain multiple boot environments on their systems, and each boot environment can have different software versions installed. Creating a backup of your existing boot environment gives you the flexibility of having an environment in which to test and validate changes as part of your troubleshooting activities instead of having to work in the production environment. After you have tested and validated your proposed fixes to a problem, you can either make the alternate boot environment the active boot environment or you can make the fixes directly in the original boot environment.

You can also manually create a snapshot of an existing boot environment for reference to facilitate your troubleshooting activities. This snapshot is a read-only image of a dataset or boot environment at a given point in time.

You also have the option of creating a new boot environment from an existing ZFS snapshot. Then you can activate and boot that new boot environment.

Alternate Boot Environments Example

To create an alternate boot environment, use the `beadm` command as seen below. Then you can activate it and it will be active on the next `reboot`:

```
# beadm create be_test
# beadm activate be_test
# beadm list
```

BE	Active	Mountpoint	Space	Policy	Created
--	----	-----	-----	-----	-----
be_test	R	-	9.89G	static	2000-02-24 13:06
solaris	N	/	51.64M	static	2000-02-08 15:17
solaris.orig	-	-	1.17M	static	2000-02-08 17:23

```
# reboot
# beadm list
```

BE	Active	Mountpoint	Space	Policy	Created
--	----	-----	-----	-----	-----
be_test	NR	-	9.89G	static	2000-02-24 13:06
solaris		/	51.64M	static	2000-02-08 15:17
solaris.orig	-	-	1.17M	static	2000-02-08 17:23

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can create multiple boot environments which may be helpful for setting up test environments or as a safeguard. You can also boot to an alternate boot environment at the PROM level by using the `boot -Z rpool/ROOT/be_test` command at the `ok` prompt.

Booting a Backup Boot Environment

- On SPARC-based systems, use then OBP command boot -L to identify your backup BEs.

```
ok> boot -L
. . .
Boot device:
    /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a File
    and args: -L
zfs-file-system Loading: /platformsun4v/bootlst
1.s11_update1
2 backup_be
Select environment to boot: [ 1 - 2 ]: 2
to boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/backup_be
ok> boot -Z rpool/ROOT/backup_be
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When booting from ZFS, the *device-specifier* identifies a storage pool, not a single root file system. A storage pool can contain multiple boot environments, datasets, or root file systems. Therefore, when booting from ZFS, you must also identify a root file system within the pool that is identified by the boot device as the default. The default boot device is identified by the pool's `bootfs` property. To identify the path of the root file system for the available BEs, use the OBP command `boot -L`.

Case Study 3: Solaris Fails to Start!

- Clarify problem statement.
 - SPARC-based system `host2`, fails to boot up.
 - The system goes in to maintenance mode.
- Using “assumptive” method, prioritize the most probable causes of failure.
 - SMF related problem
 - Faulty software configuration
- Gather pertinent information.
 - Check SMF service status.
 - Peruse system and service logs.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 3: Troubleshooting Steps

The follows steps are used to identify the cause of the problem:

1. Search `/var/adm/messages` file for error messages.
2. Check SMF services status including milestones.
3. Check SMF service logs for any services requiring maintenance.
4. If milestones are not online, reboot without milestones.
5. Determine milestone dependencies.
6. Repair failing service on which milestone is dependent.
7. Verify that the system boots normally.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the high-level steps involves in this case study.

Case Study 3: Walkthrough

```
##### Boot the Solaris OS and observer the failure. #####
ok boot
Sun Fire T200, No Keyboard
Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #67263734.
Ethernet address 0:14:4f:2:5c:f6, Host ID: 84025cf6.

Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a File and args:
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Requesting System Maintenance Mode
(See /lib/svc/share/README for more i
nformation.)
Console login service(s) cannot run

Enter user name for system maintenance (control-d to bypass): root
Password: xxxxxx

Mar 18 17:51:24 su: 'su root' succeeded for root on /dev/console
Oracle Corporation      SunOS 5.11      11.1      April 2013
root@:~#
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide begin the Case Study 3 walkthrough. Note that line beginning with “#####” are comments.

Case Study 3: Walkthrough

```
##### Check the /var/adm/messages file for error indications. #####
root@:~# cat /var/adm/messages | egrep '(panic)|(trap)|(error)'
root@:~#
##### Check the milestone status to determine how far the startup progressed. #####
root@:~# svcs "svc:/milestone/*"
STATE          STIME      FMRI
online         17:50:16  svc:/milestone/unconfig:default
online         17:50:16  svc:/milestone/config:default
online         17:50:17  svc:/milestone/devices:default
online         17:50:25  svc:/milestone/network:default
online         17:50:26  svc:/milestone/name-services:default
online         17:50:28  svc:/milestone/self-assembly-complete:default
offline        17:49:36  svc:/milestone/single-user:default
offline        17:49:44  svc:/milestone/multi-user:default
offline        17:49:49  svc:/milestone/multi-user-server:default
##### Reboot the system with no milestones. #####
root@:~# init 0
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 3: Walkthrough

```
ok boot -m milestone=none
SC Alert: Host System has Reset
Sun Fire T200, No Keyboard
Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #67263734.
Ethernet address 0:14:4f:2:5c:f6, Host ID: 84025cf6.

Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a File and ar
gs: -m milestone=none
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Booting to milestone "none".
Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run

Enter user name for system maintenance (control-d to bypass): root
Enter root password (control-d to bypass): XXXXXX
single-user privilege assigned to root on /dev/console.
Entering System Maintenance Mode
root@:~#
##### Try to manually bring the single-user milestone online. #####
root@:~# svcadm milestone svc:/milestone/single-user:default
root@:~# svcsvcs single-user
STATE          STIME          FMRI
offline        18:08:00      svc:/milestone/single-user:default
root@:~#
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 3: Walkthrough

```
##### Determine the single-user milestone dependencies. #####
root@:~# svcs -d single-user
STATE      STIME      FMRI
disabled   18:06:27   svc:/system/identity:node
online     18:08:03   svc:/system/cryptosvc:default
online     18:08:11   svc:/network/loopback:default
online     18:08:12   svc:/system/filesystem/minimal:default
online     18:08:12   svc:/network/uucp-lock-cleanup:default
online     18:08:12   svc:/system/sysevent:default
online     18:08:13   svc:/system/environment:init
online     18:08:13   svc:/milestone/unconfig:default
online     18:08:13   svc:/milestone/config:default
online     18:08:13   svc:/milestone/devices:default
online     18:08:15   svc:/system/manifest-import:default
online     18:08:15   svc:/system/config-user:default
online     18:08:15   svc:/system/timezone:default
online     18:08:15   svc:/system/keymap:default
online     18:08:21   svc:/milestone/network:default
offline    18:08:02   svc:/system/picl:default
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to its upper right.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 3: Walkthrough

```
##### Determine the grouping properties of the services on which #####
##### the single-user milestone is dependent. #####
root@:~# more /lib/svc/manifest/milestone/single-user.xml
...
    <dependency
        name='nodename'
        grouping='require_all'
        restart_on='none'
        type='service'>
        <service_fmri value='svc:/system/identity:node' />
    </dependency>
...
##### Manually enable the offline service on which single-user is dependent. #####
root@:~# svcadm enable svc:/system/identity:node
Hostname: host02
##### Try to manually bring the single-user milestone online. #####
root@:~# svcadm milestone svc:/milestone/single-user:default
root@:~# svcs single-user
STATE      STIME      FMRI
online     18:09:45   svc:/milestone/single-user:default
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 3: Walkthrough

```
##### Try to manually bring the multi-user milestone online. #####
root@:~# svcadm milestone svc:/milestone/multi-user:default
root@:~# svcs multi-user
STATE          STIME      FMRI
online          18:10:52  svc:/milestone/multi-user:default
##### Try to manually bring the multi-user-server milestone online. #####
root@:~# svcadm milestone svc:/milestone/multi-user-server:default
root@:~# svcs multi-user-server
STATE          STIME      FMRI
online          18:11:31  svc:/milestone/multi-user-server:default
##### Reboot and verify that the system does not enter maintenance mode. #####
root@:~# reboot
...

Sun Fire T200, No Keyboard
Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #67263734.
Ethernet address 0:14:4f:2:5c:f6, Host ID: 84025cf6.

Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a  File and args:
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Hostname: host02

host02 console login:
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 4: Overview

- 4-1: Restoring from a back-up system file
- 4-2: Restoring corrupt boot program files
- 4-3: Restoring from an alternate boot device
- 4-4: Debugging an SMF service that is in a maintenance state

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Identify the stages associated with a successful Solaris OS startup
- List faults common to Solaris OS startup failures
- Identify the best method for troubleshooting Solaris OS startup failure
- Determine best tools for analyzing and troubleshooting Solaris OS startup failures
- Identify the best tools and method for analyzing and troubleshooting failures associated with system services
- Determine when and how to use Solaris Boot Environments (BEs) for quick operating system recovery

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

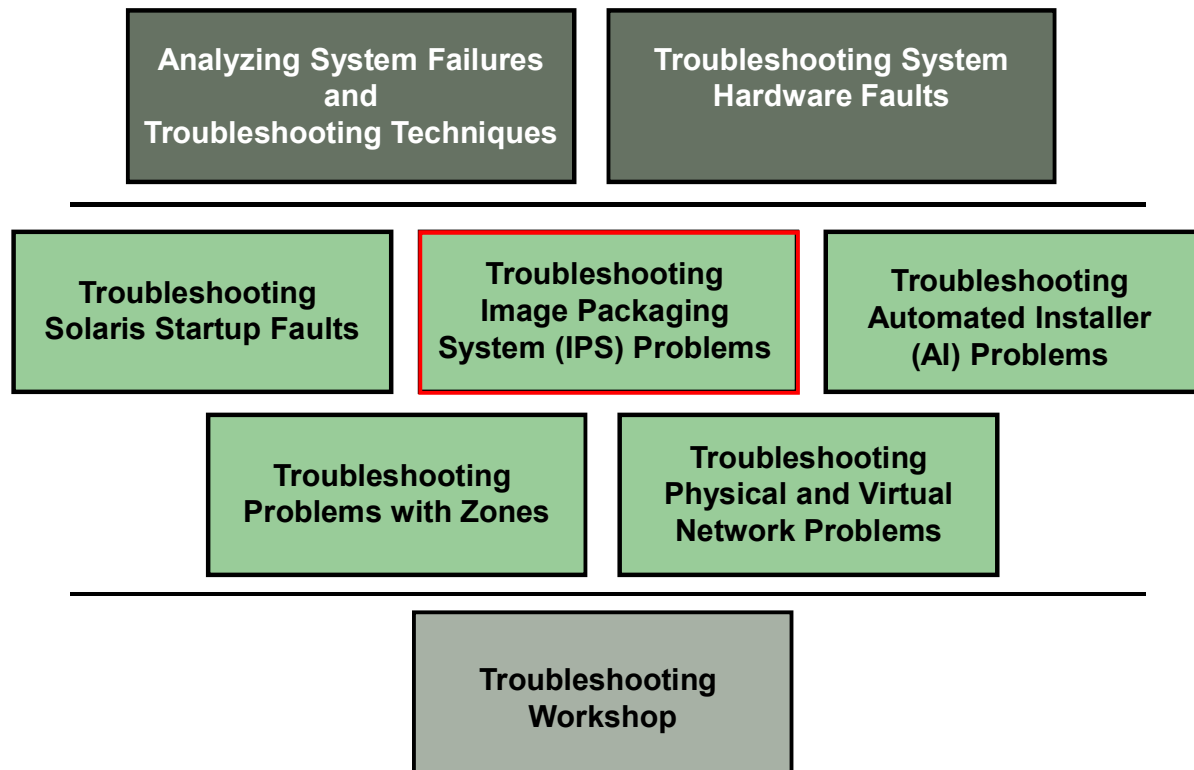
5

Troubleshooting Image Packaging System (IPS) Problems

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this lesson, Troubleshooting Image Packaging System (IPS) Problems, we focus on potential IPS issues, their causes, and how to troubleshoot and resolve these issues.

Objectives

After completing this lesson, you should be able to:

- Describe how the Image Packaging System (IPS) functions
- Verify the IPS configuration
- Install software packages
- Update software packages
- Fix package problems

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

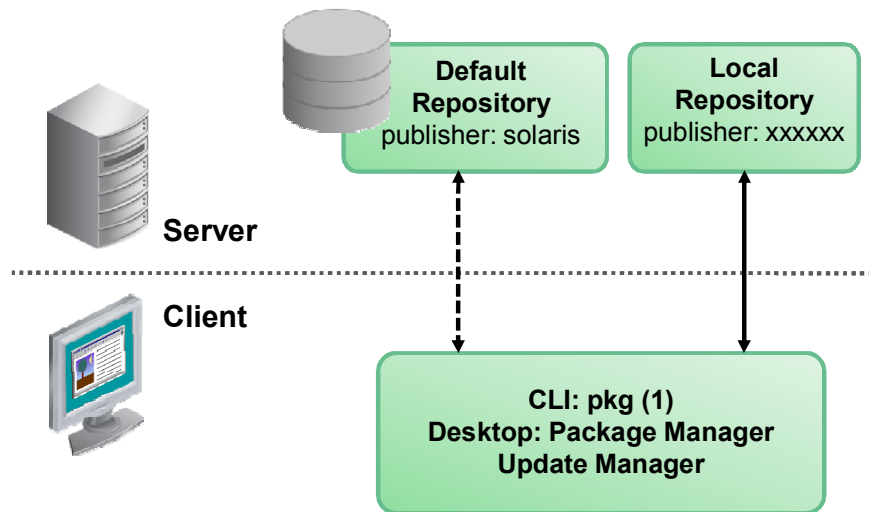
Agenda

- Reviewing IPS Features and Functionality
- Checking the Basics
- Installing and Updating Packages
- Fixing Package Problems
- Case Study 4

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Reviewing the IPS Features



ORACLE

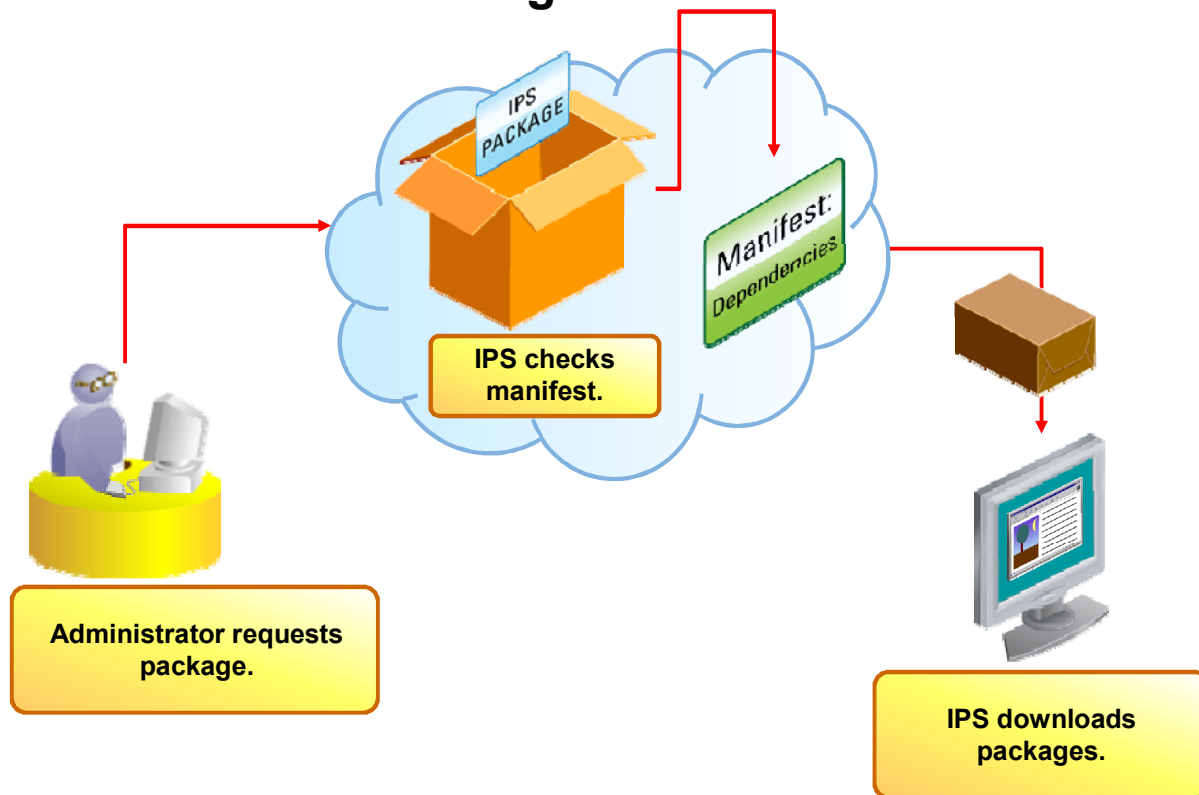
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

You can administer packages by using the `pkg` command from the command line interface, the Package Manager from the desktop GUI, or Update Manager.

Reviewing How IPS Works



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

After IPS has installed the packages to the client system, you can manage them. You can list, search, and uninstall packages as well as perform a variety of other functions.

In a console environment, package updates are performed using the package update command (`pkg update`). During the update process, each package is updated by default from the publisher that provided the current installed version.

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

Agenda

- Reviewing IPS Features and Functionality
- **Checking the Basics**
- Installing and Updating Packages
- Fixing Package Problems
- Case Study 4

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Example Problem

```
Downloading linked: 0/1 done; 1 working: zone:web  
Linked progress: /  
pkg: An unexpected error happened while preparing for update:  
pkg: Linked image exception(s):
```

A 'update' operation failed for child 'zone:web' with an unexpected return value of 1 and generated the following output:

```
pkg: The plan for the current operation is no longer valid.  
The image has likely been modified by another process or  
client. Please try the operation again.
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here you see an error message caused by referencing a stale (out-of-date) local publisher when performing a package update operation.

Determining the Client Host and Domain Names

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

```
# hostname  
host01  
  
# domainname  
mydomain.com
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For purposes of training, let's assume we are dealing with a local IPS server. 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 server
Server:          192.168.0.100
Address:         192.168.0.100#53

Name:           server.mydomain.com
Address:        192.168.0.100

# ping server
server is alive
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Next, you want to verify that the client machine can access DNS services and that it can connect to the local IPS server.

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

Verifying the Local IPS Publisher Setting

Check the current publisher. If it is not correct, use the `pkg set-publisher` command to set the publisher to the local IPS repository.

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

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

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

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

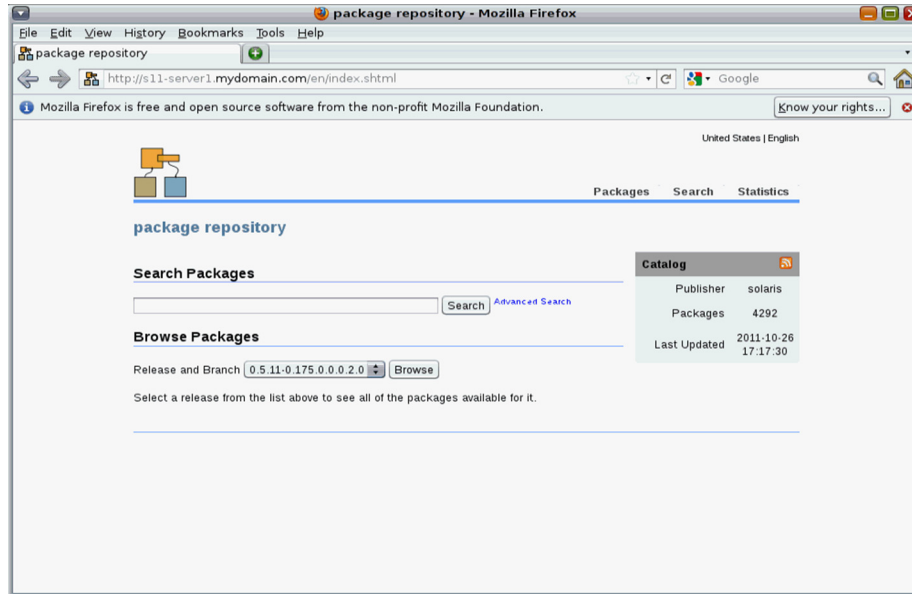
After you have verified network connectivity between the client and the local IPS server, you need to verify that the local IPS publisher has been set correctly.

First, check the current publisher. If the publisher is not set correctly, set it to the local IPS repository, and then verify that the publisher is now the local IPS publisher.

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

To test client access to the IPS server, open the local publisher URI in a browser.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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.

Note: In this example, the URI would be `http://server.mydomain.com`.

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
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There may be times when you need to disable a publisher temporarily. 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.

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 Origin URI

To change a publisher 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
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There may be times when you want or need to change a publisher's origin URI. 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.

Agenda

- Reviewing IPS Features and Functionality
- Checking the Basics
- **Installing and Updating Packages**
- Fixing Package Problems
- Case Study 4

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Previewing a Package Installation Operation

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

```
# pkg install -nv appttrace
      Packages to install:           1
      Estimated space available:     46.27 GB
      Estimated space to be consumed: 13.55 MB
      Create boot environment:       No
      Create backup boot environment: No
      Rebuild boot archive:          No
...
...
...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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 this example, we are checking into the details of installing the `appttrace` package.

Once you are confident that there will be no issue with installing the package, you can run the `pkg install` command to complete the package installation operation.

Verifying Package Installation

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

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

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After the package installation operation has completed, you can validate or verify the installation of the package by using the `pkg verify` command.

Note

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

If you do not use the `-v` option, there is no output if the packages are validated successfully.

Rejecting a Package

To reject a package, use the `--reject` option with `pkg install` or `pkg update`.

```
# pkg update --reject OpenOffice
```

The `--reject` option:

- Prevents the specified packages from being added to the system
- Removes the specified packages from the system if they are already installed
- Allows uninstall and update operations in one command
- Is useful for dealing with dependency knots

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Oracle Solaris 11, you now have the ability to prevent certain packages from being installed by using the `--reject` command with the `pkg install` and `pkg update` commands. As part of this command, you specify a package name pattern, as shown in the example in the slide. When you run the command, any package names that match the specified `pkg_fmri_pattern` are prevented from being installed. If matching packages are already installed, they are removed as part of this operation.

For instance, as shown in the example in the slide, if you run `pkg update --reject OpenOffice`, the OpenOffice package will not be installed on the system you are updating and if it already exists on the system, it will be removed. This is a big time saver for uninstalling and updating in one command, especially if you cannot update to a certain level of software because of a previous existing dependency.

Using the Group Dependency and Avoid List

- Group dependency: Indicates to install this package unless it is on the avoid list
- A package is put on the avoid list when a:
 - User uninstalls the package and the package is currently the target of a group dependency
 - User uses the `--reject` option on the package and the package is currently the target of a group dependency
 - User uses `pkg avoid` to place the package on the avoid list
- To remove a package from the avoid list, use `pkg unavoid`.
- To view packages currently on the avoid list, use `pkg avoid` or `unavoid` with no arguments

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Also new with Oracle Solaris 11 is the group dependency and avoid list feature. A group dependency means all the packages in the group install, unless a package has been specified not to, and then it goes on the avoid list. A package can appear on the avoid list in a number of ways:

- The package has been uninstalled and it is part of a group dependency
- The `--reject` option has been used on the package and the package is part of a group dependency
- The package has been purposely placed on the avoid list by using the `pkg avoid` command.

You can remove a package from the avoid list by using the `pkg unavoid` command. You can use the `pkg avoid` command or `pkg unavoid` command with no arguments to see which packages are currently on the avoid list.

Requesting the Newest Version of a Package

To request the newest version of a package, run `pkg update *@latest`.

```
# pkg update appttrace@latest
```

The `pkg update *@latest` command:

- Forces `pkg` to try to move every installed package to its newest version
- Is useful when the `pkg update` command does not provide the expected results
- Includes new and improved error messaging to facilitate troubleshooting

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

As you know, to update an installed package in Oracle Solaris 11, you can use either `pkg install` or `pkg update`.

Note: If you want to avoid unintentionally installing a package that was not already installed, use the `pkg update` command to update packages.

In Oracle Solaris 11, you can explicitly request the newest version of a package by running the `pkg update pkg_fmri_pattern@latest` command.

In the example shown in the slide, we have requested the latest version of the `appttrace` package.

This command is useful when the `pkg update` command does not provide the expected results, such as when the initial update operation is rejected.

In addition, the `pkg update *@latest` command includes new and improved error messaging to facilitate troubleshooting, an example of which is shown on the next two slides.

Improved Error Message Example #1

```
pkg update: No matching version of entire can be installed:
  Reject: pkg://solaris/entire@0.5.11,5.11-0.171:20110801T164723Z
  Reason: All versions matching 'incorporate' dependency
pkg:/consolidation/ips/ips-incorporation@0.5.11,5.11-0.171 are rejected
  Reject: pkg://solaris/consolidation/ips/ips-incorporation@0.5.11,5.11-
0.171:20110801T133422Z
  Reason: All versions matching 'incorporate' dependency
pkg:/group/system/solaris-desktop@0.5.11,5.11-0.171 are rejected
  Reject: pkg://solaris/group/system/solaris-desktop@0.5.11,5.11-
0.171:20110801T133450Z
  Reason: All versions matching 'group' dependency
pkg:/system/library/dbus/dbus-x11 are rejected
  Reject: pkg://solaris/system/library/dbus/dbus-x11@1.2.28,5.11-
0.170:20110719T171647Z
  Reason: Excluded by proposed incorporation 'consolidation/gnome/gnome-
incorporation'
      Newer version pkg://solaris/system/library/dbus/dbus-
x11@1.2.28,5.11-0.173.0.0.0.0:20110828T052643Z is already installed
      Reject: pkg://solaris/system/library/dbus/dbus-x11@1.2.28,5.11-
0.171:20110801T074138Z
      Reason: Newer version pkg://solaris/system/library/dbus/dbus-
x11@1.2.28,5.11-0.173.0.0.0.0:20110828T052643Z is already installed
      Reject: pkg://solaris/system/library/dbus/dbus-x11@1.2.28,5.11-
0.173.0.0.0.0:20110828T052643Z
      Reason: Excluded by proposed incorporation 'consolidation/gnome/gnome-
incorporation'
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the improved error messaging, each level of indentation provides more detail about the previous reason as to why the package update failed. Often the most helpful information is the most indented, so that is a good starting place for troubleshooting. In the example presented in the slide, the most important information is presented in bolded text. In the second line, we see that the user is trying to update to build 171 of Solaris but on the tenth line, which is the most indented, shows that `dbus` from build 173 has already been installed, which is a new version.

Improved Error Message Example #2

```

No suitable version of required package
pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-0.152:20101113T031600Z found:
  Reject: pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-0.152:20101113T031600Z
  Reason: All acceptable versions of 'require' dependency on
pkg:/consolidation/hcts/hcts-incorporation are obsolete
No suitable version of required package
pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-0.151:20101027T141559Z found:
  Reject: pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-0.151:20101027T141559Z
  Reason: All acceptable versions of 'require' dependency on
pkg:/consolidation/hcts/hcts-incorporation are obsolete
No suitable version of required package
pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-0.151.0.1:20101105T053905Z found:
  Reject: pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-
0.151.0.1:20101105T053905Z
  Reason: All acceptable versions of 'require' dependency on
pkg:/consolidation/hcts/hcts-incorporation are obsolete
No suitable version of required package
pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-0.153:20101119T062300Z found:
  Reject: pkg://solaris/diagnostic/ddu/text@0.5.11,5.11-0.153:20101119T062300Z
  Reason: All acceptable versions of 'require' dependency on
pkg:/consolidation/hcts/hcts-incorporation are obsolete

```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in this slide, the user was attempting to update from build 172 to build 173 and is receiving numerous errors about the `ddu/text` package because of a dependency on the `hcts-incorporation` package. The next step is for the user to uninstall the `hcts-incorporation` package or use the new `--reject` option on that package on the next package update attempt.

Improved Error Message Example #3

```
Dependency analysis is unable to determine exact cause.  
Try specifying expected results to obtain more detailed error messages
```

```
# pkg update --reject solaris/diagnostic/ddu/text \  
--reject consolidation/hcts/hcts-incorporation *@latest
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in this slide, the user uninstalled the `hcts-incorporation` package but still received this error message when attempting to run the package update command. At this point, the user can use the `pkg update *@latest` command to obtain a more descriptive error message or include the `--reject` option in conjunction with the `*@latest` command to remove the two problematic packages and force an update of all other packages as far forward as possible, thereby allowing the update to proceed. You can see an example of this second approach in the second example in the slide.

Backing Out Packages

To downgrade an installed package to a previous version, run `pkg update pkg_fmri@version`.

```
# pkg update appttrace@5.10
```

To remove an installed package, run `pkg uninstall pkg_fmri_pattern`.

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

Planning linked: 0/3 done; 1 working: zone:database
Planning linked: 1/3 done; 1 working: zone:web
Planning linked: 2/3 done; 1 working: zone:storage
Planning linked: 3/3 done
<output omitted>
#
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There might be times when you need to back out an installed package. If you just want to revert to an older version of the currently installed package, you can perform an in-place downgrade by using the `pkg update pkg_fmri@version` command, as shown in the first example in the slide.

Note

Any preserved configuration files that are part of packages to be downgraded and that have been changed since the original version was installed are renamed with the extension `.update`.

On the other hand, if you want to remove a package or packages completely, you can use the `pkg uninstall` command, as shown in the second example in the slide.

When you are finished with all install, update, and uninstall operations, it's a good idea to use `pkg refresh` to update the list of available packages and publisher metadata for each publisher specified. If no publishers are specified, the refresh is performed for all publishers.

For More Information

- `pkg` man page
- External mailing list:
`pkg-discuss@opensolaris.org`
- Internal mailing list:
`pkg-core_ww@oracle.com`
- To file bugs:
https://defect.opensolaris.org/bz/enter_bug.cgi?product=pkg
- To find bugs:
<https://defect.opensolaris.org/bz/query.cgi>
Choose product `pkg`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For more information about IPS, consult the `pkg` man page and use the appropriate mailing list. Note also the URLs for filing and finding bugs.

Quiz

The `--reject` command when used with `pkg install` or `pkg update` prevents certain packages from being installed, but it does not remove matching packages that have already been installed.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

In Oracle Solaris 11, you can explicitly request the newest version of a package by running the `pkg update pkg_fmri_pattern@latest` command.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Agenda

- Reviewing IPS Features and Functionality
- Checking the Basics
- Installing and Updating Packages
- **Fixing Package Problems**
- Case Study 4

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Checking the History Log

To check the history log of all package operations, run `pkg history`.

```
# pkg history
START                OPERATION                CLIENT                OUTCOME
2013-04-02T19:39:13  set-property            transfer module       Succeeded
2013-04-02T19:39:13  image-create            transfer module       Succeeded
2013-04-02T19:39:14  add-publisher           transfer module       Succeeded
2013-04-02T19:39:14  refresh-publishers      transfer module       Succeeded
2013-04-02T19:39:23  rebuild-image-catalogs  transfer module       Succeeded
2013-04-02T19:39:31  add-publisher           transfer module       Succeeded
2013-04-02T19:39:31  refresh-publishers      transfer module       Succeeded
2013-04-02T19:39:32  rebuild-image-catalogs  transfer module       Succeeded
2013-04-02T19:39:40  install                 transfer module       Succeeded
2013-04-02T21:41:05  uninstall               pkg                   Succeeded
2013-04-18T10:54:34  install                 pkg                   Succeeded
2013-04-18T11:29:56  update                  pkg                   Failed
2013-04-18T11:37:35  update                  pkg                   Failed
2013-04-18T11:38:12  update                  pkg                   Failed
2013-04-18T11:38:51  uninstall               pkg                   Succeeded
#
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can check the history log of all package operations by using the `pkg history` command.

The output (as shown in the example in the slide) contains the date and time the operation began, what the operation was, the client that was performing the operation, and the outcome.

In this example, you see that we have three failed package update operations.

For a complete list of options you can use with the `pkg history` command, see the Image Packaging System Man Pages `pkg(1)`.

Checking the History Log

To search the history log by set of date stamps, run `pkg history -t date_stamp-date_stamp`.

```
# pkg history -t 2013-04-02T19:39:13-2013-04-02T19:39:14
```

START	OPERATION	CLIENT	OUTCOME
2013-04-02T19:39:13	set-property	transfer module	Succeeded
2013-04-02T19:39:13	image-create	transfer module	Succeeded
2013-04-02T19:39:14	add-publisher	transfer module	Succeeded
2013-04-02T19:39:14	refresh-publishers	transfer module	Succeeded



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can also search the history log by using the `pkg history` command with the `-t` option and a set of date stamps, as shown in the example in the slide.

In this example, we are searching for package operations between the dates of 2013-04-02T19:39:13 and 2013-04-02T19:39:14.

Checking the History Log

To select different output options, run `pkg history -o column_header,column_header`.

```
# pkg history -o user,be,time,command -t 2013-04-18T11:29:56-2013-04-18T11:38:12
USER      BE      TIME      COMMAND
root      solaris  0:00:09   /usr/bin/pkg update aptrace@latest
root      solaris  0:00:08   /usr/bin/pkg update apptrace@5.10
root      solaris  0:00:09   /usr/bin/pkg update apptrace@0.5.10
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You also have the option of displaying different output by using the `pkg history -o` option to select different column headers. For example, you could display the user who started the operation (`user`), the boot environment the operation was applied to (`be`), the time it took for the packaging operation to complete (`time`), and the command used to run the operation.

In the example shown in the slide, we have selected to view the user, boot environment, time, and command information for the three failed package update operations that we saw when we first ran the `pkg history` command (see slide 33).

Determining the Package Payload

To identify the content a package, run the `pkg contents` command:

```
# pkg contents e1000g
PATH
kernel
kernel/drv
kernel/drv/e1000g.conf
kernel/drv/sparcv9
kernel/drv/sparcv9/e1000g
usr/share/man/man7d
usr/share/man/man7d/e1000.7d
usr/share/man/man7d/e1000g.7d
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you are having problems with a software feature or device such as corruption or missing components, it is important to identify the contents of the package used to install that software feature or device driver. This example shows the `e1000g` package directories and files (payload).

Determining Package Actions and Dependencies

To identify the actions and dependencies of a package, run the `pkg contents -rm` command:

```
# pkg contents -rm e1000g
set name=pkg.fmri value=pkg://solaris/driver/network/e1000g@0.5.11,5.11-
0.173.0.0.0.1.0:20110826T151051Z
set name=variant.arch value=sparc value=i386
set name=org.opensolaris.consolidation value=osnet
set name=variant.opensolaris.zone value=global value=nonglobal
set name=pkg.fmri
value=pkg://solaris/driver/network/ethernet/e1000g@0.5.11,5.11-
0.175.1.0.0.24.2:20120919T184308Z
set name=pkg.summary value="Intel(R) PRO/1000 NIC Driver"
set name=pkg.description value="Network device driver for the Intel(R)
PRO/1000 Server Adapter family of Gigabit Ethernet NICs."
...
depend fmri=consolidation/osnet/osnet-incorporation type=require
depend fmri=pkg://driver/network/ethernet/e1000g@0.5.11,5.11-
0.173.0.0.0.0.0 type=require
depend fmri=feature/package/dependency/self type=parent
variant.opensolaris.zone=nonglobal
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In addition to the payload, it is helpful to identify the actions performed by the package and the package dependencies. Note that the `-r` option displays information for the newest available versions of packages that could be installed in this image from the repositories of the publishers configured in this image. The `-m` option displays all attributes of all actions in the specified package.

Verifying Package Installation

```
# pkg verify -v system/install/auto-install/auto-install-common
PACKAGE                                STATUS
pkg://solaris/system/install/auto-install/auto-install-common  ERROR
    file: usr/share/auto_install/manifest/default.xml
    Missing: regular file does not exist
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the last section on Installing and Updating Packages, you saw how to verify or validate a package installation by using the `pkg verify` command. You might also want to use this same command as part of diagnosing the cause of a package problem.

In this example, we are verifying an auto-install-common package. Let's assume that modifications have been made to the package and we are seeing errors. When we run the verification, as shown in this example, you can see that IPS has detected a missing file error (`default.xml`). In the next slide, you are shown how to fix the problem.

Fixing Verification Errors

To fix a verification error, run `pkg fix pkg-fmri`.

```
# pkg fix --accept system/install/auto-install/auto-install-common
Verifying: pkg://solaris/system/install/auto-install/auto-install-common
ERROR
    file: usr/share/auto_install/manifest/default.xml
    Missing: regular file does not exist
Created ZFS snapshot: 2011-09-28-05:34:02
Repairing: pkg://solaris/system/install/auto-install/auto-install-common

DOWNLOAD                                PKGS      FILES      XFER (MB)
Completed                               1/1        1/1        0.0/0.0

PHASE                                ACTIONS
Update Phase                           1/1

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

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can fix package verification errors using the `pkg fix` command, as shown in the slide.

Use the `--accept` option to indicate that you agree to and accept the terms of the licenses of the packages that are updated or installed. If you do not provide this option, and any package licenses require acceptance, the fix operation fails. Use the `--licenses` option to display all of the licenses for the packages that are updated as part of this operation.

Note: A ZFS snapshot is created as part of the `pkg fix` command.

Restoring a File in a Package

To restore a file in a package, run `pkg revert pkg-fmri`.

```
# pkg revert /usr/share/auto_install/sc_profiles/static_network.xml
    Packages to update: 1
    Create boot environment: No
    Create backup boot environment: No

DOWNLOAD                                PKGS          FILES      XFER (MB)
  SPEED
Completed                               1/1           1/1        0.0/0.0
  0B/s

PHASE                                ITEMS
Updating modified actions              1/1
Updating image state                   Done
Creating fast lookup database          Done
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If a file or files in a package have become corrupted, you can restore the file or files to their as-delivered condition by using the `pkg revert` command. When you revert the file or files, file ownership and protections are also restored.

Caution: Reverting some editable files to their default values can make the system unbootable, or cause other malfunctions.

Quiz

Assume you run the `pkg verify` command to validate the installation of a package called `wireshark` and a number of errors are reported. Which command can you use to correct the package installation errors?

- a. `pkg fix wireshark`
- b. `pkg update --reject wireshark`
- c. `pkg revert wireshark`
- d. `pkg uninstall wireshark`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Agenda

- Reviewing IPS Features and Functionality
- Checking the Basics
- Installing and Updating Packages
- Fixing Package Problems
- **Case Study 4**

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 4: DTrace Not Working Properly!

- Clarify problem statement.
 - DTrace is unable to monitor system locks on `host02`!
- Using “assumptive” method, prioritize the most probable causes of failure.
 - Problem with DTrace software
- Starting with the most probable cause of failure, gather pertinent information.
 - Check related software package

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 4: Troubleshooting Steps

Perform the following steps to identify the cause of the problem:

1. Verify that DTrace is not functioning properly.
2. Check the DTrace software package installation.
3. Verify that the failing software component is bundled in the DTrace software package.
4. Repair the fault.
5. Verify the solution.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the high-level steps involved in this case study.

Case Study 4: Walkthrough

```
##### Verify that DTrace is not functioning properly. #####
root@host02:~# dtrace -n 'lockstat::adaptive-block \
    { @time[execname] = sum(arg1); }'
dtrace: invalid probe specifier lockstat::adaptive-block { @time[execname] =
    sum(arg1); }: probe description lockstat::adaptive-block does not match any
    probes

root@host02:~# dtrace -l | perl -pe 's/^\.*?\s+\s+(\s+?) ([0-9] |\s) .*/\1/' | sort \
    | uniq | grep lockstat
root@host02:~#
##### Note that the lockstat probe is not listed. #####
##### Verify the integrity of the DTrace software package. #####
root@host02:~# pkg verify dtrace
PACKAGE                                STATUS
pkg://solaris/system/dtrace            ERROR
    file: kernel/drv/sparcv9/lockstat
        Missing: regular file does not exist
    hardlink: kernel/dtrace/sparcv9/lockstat
        Target '../kernel/drv/sparcv9/lockstat' does not exist
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide begin the Case Study 4 walkthrough. Note that line beginning with “#####” are comments.

Case Study 4: Walkthrough

```
##### Verify that lockstat is comes with the DTrace software package. #####
root@host02:~# pkg contents dtrace |grep lockstat
kernel/drv/lockstat.conf
kernel/drv/sparcv9/lockstat
kernel/dtrace/sparcv9/lockstat
usr/include/sys/lockstat.h
usr/sbin/lockstat
usr/sbin/plockstat
usr/sbin/sparcv9/lockstat
usr/sbin/sparcv9/plockstat
usr/share/man/ja_JP.UTF-8/man1m/plockstat.1m
usr/share/man/man1m/lockstat.1m
usr/share/man/man1m/plockstat.1m
usr/share/man/man7d/lockstat.7d
usr/share/man/zh_CN.UTF-8/man1m/plockstat.1m
##### Try to repair the DTrace software package. #####
root@host02:~# pkg fix dtrace
Verifying: pkg://solaris/system/dtrace                                ERROR
      file: kernel/drv/sparcv9/lockstat
            Missing: regular file does not exist
      hardlink: kernel/dtrace/sparcv9/lockstat
            Target '../kernel/drv/sparcv9/lockstat' does not exist
Created ZFS snapshot: 2000-02-01-01:19:15
Repairing: pkg://solaris/system/dtrace
Creating Plan (Evaluating mediators): -pkg: Requested "fix" operation would affect files
      that cannot be modified in live image.
Please retry this operation on an alternate boot environment.
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 4: Walkthrough

```
##### Reboot from and alternate BE.#####
root@host02:~# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      NR      /           9.85G  static 2013-03-19 19:07
solaris.orig -      -           1.12M  static 2013-03-19 21:28
root@host02:~# beadm activate solaris.orig
root@host02:~# reboot
...
host02 console login: root
Password: XXXXXX
Mar 19 23:32:03 host02 login: ROOT LOGIN /dev/console
Last login: Tue Mar 19 23:04:20 on console
Oracle Corporation      SunOS 5.11      11.1      April 2013
##### Mount the original BE. #####
root@host02:~# beadm mount solaris /mnt
root@host02:~# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      -      /mnt       207.95M static 2000-01-22 20:32
solaris.orig NR      /           9.64G  static 2000-01-22 22:52
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 4: Walkthrough

```
##### Repair the DTrace software package. #####
root@host02:~# pkg -R /mnt fix dtrace
Verifying: pkg://solaris/system/dtrace
           file: kernel/drv/sparcv9/lockstat
           Missing: regular file does not exist
           hardlink: kernel/dtrace/sparcv9/lockstat
           Target '../kernel/drv/sparcv9/lockstat' does not exist
Created ZFS snapshot: 2000-02-01-01:28:40
Repairing: pkg://solaris/system/dtrace
Creating Plan (Evaluating mediators): /

DOWNLOAD          PKGS          FILES    XFER (MB)   SPEED
Completed          1/1           1/1      0.0/0.0     0B/s

PHASE              ITEMS
Updating modified actions  3/3
Updating image state       Done
Creating fast lookup database Done
root@host02:~# beadm unmount solaris
root@host02:~# beadm activate solaris
root@host02:~# reboot
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 4: Walkthrough

```
##### Verify the DTrace repair. #####
host02 console login: root
Password: XXXXXX
Mar 19 23:13:40 host02 login: ROOT LOGIN /dev/console
Last login: Tue Mar 19 22:36:21 on console
Oracle Corporation      SunOS 5.11      11.1      April 2013
root@host02:~# dtrace -l | perl -pe 's/^\.*?\S+\s+(\S+?) ([0-9]|\s).*\/\1/' | sort \
| uniq | grep lockstat
lockstat
root@host02:~# dtrace -n 'lockstat:::adaptive-block \
{ @time[execname] = sum(arg1); }'
dtrace: description 'lockstat:::adaptive-block ' matched 1 probe
^C

      sched                                34660
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Additional Resources

- IPS Cheat Sheet:
<http://www.oracle.com/technetwork/server-storage/solaris11/documentation/ips-one-liners-032011-337775.pdf>
- OTN IPS Page (white papers, blogs, etc.):
<http://www.oracle.com/technetwork/server-storage/solaris11/technologies/ips-323421.html>
- Documentation: Adding and Updating Oracle Solaris 11.1 Software Packages:
http://docs.oracle.com/cd/E26502_01/html/E28984/index.html

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 5: Overview

- 5-1: Troubleshooting IPS problems

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Describe how the Image Packaging System (IPS) functions
- Verify the IPS configuration
- Install software packages
- Update software packages
- Fix package problems

ORACLE

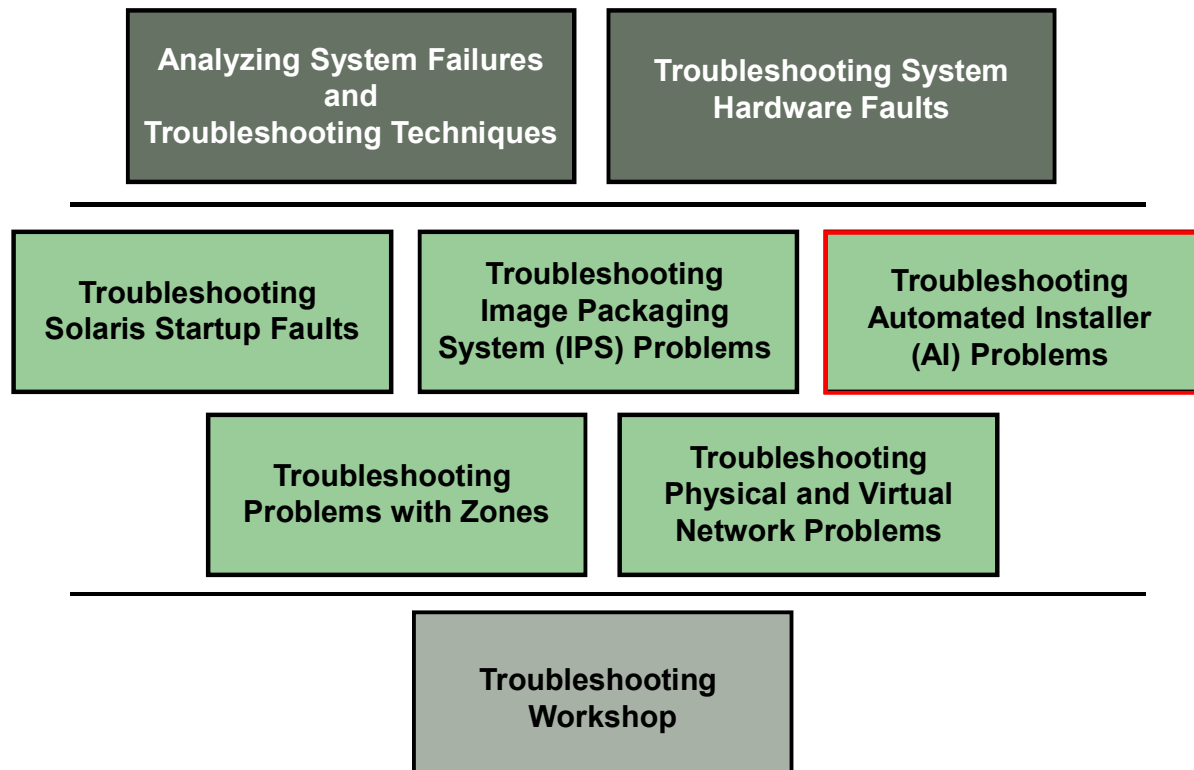
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Troubleshooting Automated Installer (AI) Problems

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this lesson, Troubleshooting Automated Installer (AI) Problems, we focus on potential AI issues, their causes, and how to troubleshoot and resolve these issues.

Objectives

After completing this lesson, you should be able to:

- Recall the Automated Installer (AI) features and functions
- Troubleshoot client installation failures

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Reviewing AI Features and Functionality
- Troubleshooting Client Installation Failures
- Case Study 5

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Automated Installation Overview

- The AI automates the installation of the OS on one or more SPARC and x86 systems over a network.
- The installations differ in architecture, software packages, disk capacity, network configuration, and other parameters.
- Automated installation requires the following components:
 - **AI server:** Provides the install service that contains the installation instructions for the client system
 - **Client system to be installed:** Accesses the IP address information from the DHCP server
 - **DHCP server:** Provides the initial IP addresses and boot information
 - **IPS repository:** Provides the software packages that are identified in the AI manifest file to the client system

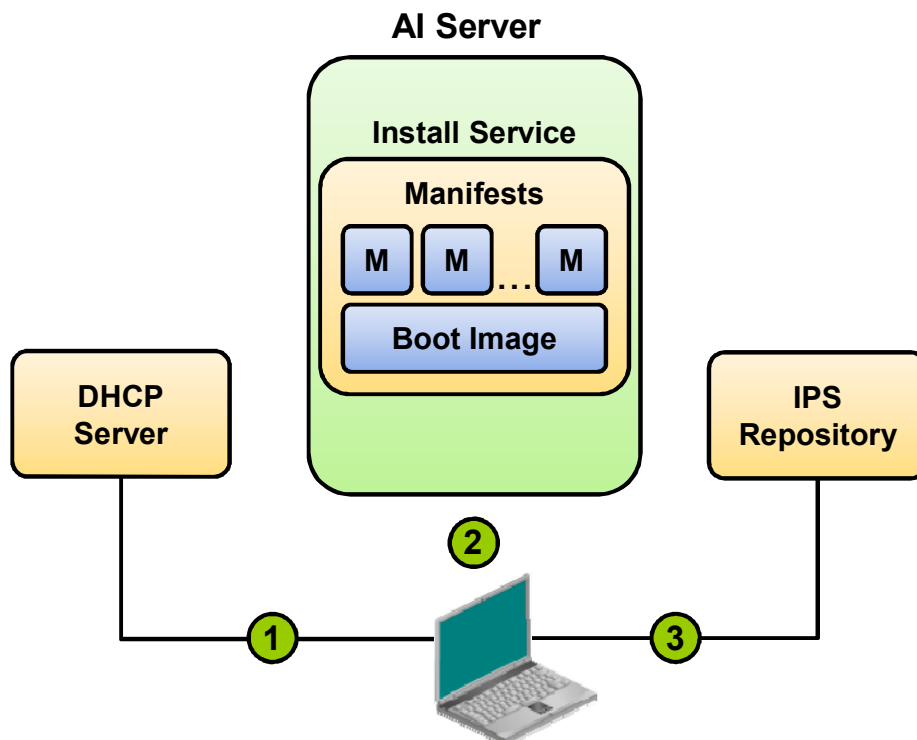
The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Automated Installer is used to automate the installation of the Oracle Solaris 11 OS on one or more SPARC and x86 systems over a network. The installations can differ in architecture, software packages installed, disk capacity, network configuration, and other parameters.

For an automated installation to run, the components presented in the slide are required.

Reviewing the Automated Installation Process



ORACLE

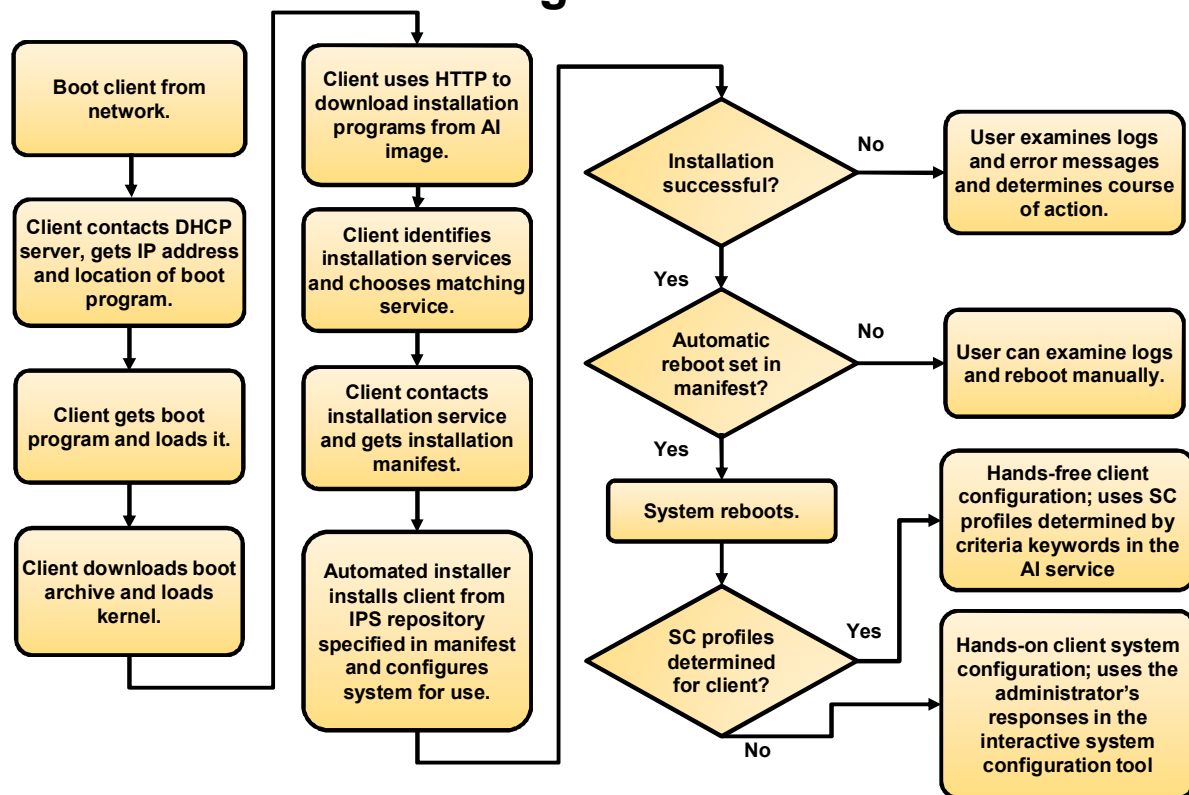
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The graphic in the slide illustrates the automated installation process:

1. A client system boots and gets IP and boot information from the DHCP server.
2. The client contacts an install service on the AI server, accesses the boot image, downloads the boot archive, and loads the kernel. It uses HTTP to download the install programs from the AI image, identifies install services, and chooses a matching service.
3. The client then contacts that install service and gets a manifest. It installs the operating system, pulling packages from the required IPS repository as specified in the corresponding manifest.

The system reboots if specified for automatic reboot in the manifest, or it can be booted manually. If the installation is not successful, you should examine the log files and error messages to determine the cause of the failure. You are shown how to troubleshoot client installation failures in the next topic.

Reviewing How AI Works



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The flowchart in the slide illustrates how a client system is installed. The client browses for available installation services, seeking a service where the installation criteria in the service's manifest file matches the characteristics of the client system. When a match is found, the installation is performed on the client system by using a boot image, and the manifest and SC profile specifications provided by the install service.

AI Components

- The system manifest files (for global and non-global zones)
- Client criteria
- The system configuration (SC) profile files (for global and non-global zones)

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

AI contains three important components that are normally created and managed by system administrators. These are:

- The system manifest files (for global and non-global zones)
- Client criteria
- The system configuration (SC) profile files (for global and non-global zones)

The *manifest* file contains a AI installation instructions such as IPS server location, software packages to be installed, and the location of a non-global zone configuration file. The *client criteria* allows you to identify which clients will be associated with specific AI services. The *SC profile* file contains information that allows you to configure client's attributes such as user accounts, networking configuration, localization, and so on.

AI Manifest Excerpts

```
...
<source>
  <publisher name="solaris">
    <origin name="http://s11-ss.mydomain.com"/>
  </publisher>
</source>
...
<software_data action="install">
  <name>pkg:/entire@latest</name>
  <name>pkg:/group/system/solaris-small-server</name>
</software_data>
</software>
  <configuration type="zone" name="engineering"
source="http://web.mydomain.com/zone_configs/engineering.cfg"/>
</ai_instance>
</auto_install>
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This example shows excerpts from an AI manifest file. Here you see the location of the IPS publisher, the software packages to be installed, and the location of a non-global zone configuration file.

AI Client Criteria Examples

- arch criteria manifest file:

```
<ai_criteria_manifest>
  <ai_criteria name="arch">
    <value>i86pc</value>
  </ai_criteria>
</ai_criteria_manifest>
```

- mac criteria manifest file:

```
<ai_criteria_manifest>
  <ai_criteria name="mac">
    <value>0:14:4F:20:53:94</value>
  </ai_criteria>
</ai_criteria_manifest>
```

- ipv4 criteria manifest file:

```
<ai_criteria_manifest>
  <ai_criteria name="ipv4">
    <value>192.168.0.114</value>
  </ai_criteria>
</ai_criteria_manifest>
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows examples of arch, mac, and ipv4 criteria files.

AI SC Profile Excerpts

```
...
<property_group type="application" name="root_account">
  <propval type="astring" name="login" value="root"/>
  <propval type="astring" name="password"
value="$5$XZO25MJN$Ar.iy6Xk/TcZdlK82mACTMymYs7tAGI8EWW2qUdJC72"/>
  <propval type="astring" name="type" value="role"/>
...
<property_group type="application" name="install_ipv4_interface">
  <propval type="astring" name="address_type" value="static"/>
  <propval type="net_address_v4" name="static_address"
value="192.168.0.113/24"/>
  <propval type="astring" name="name" value="net0/v4"/>
...
<service version="1" type="service" name="network/dns/client">
  <property_group type="application" name="config">
    <property type="net_address" name="nameserver">
      <net_address_list>
        <value_node value="192.168.0.100"/>
      </net_address_list>
    </property>
    <property type="astring" name="search">
      <astring_list>
        <value_node value="mydomain.com"/>
      </astring_list>
    </property>
  </property_group>
</service>
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This example shows excerpts from an AI manifest file. Here you see the location of the IPS publisher, the software packages to be installed, and the location of a non-global zone configuration file.

Agenda

- Reviewing AI Features and Functionality
- Troubleshooting Client Installation Failures
- Case Study 5

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Checking Logs and Other Files

This section covers the following topics:

- Gathering AI client logs and other files
- Gathering AI server logs and other files
- Using additional debugging tools

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Gathering AI Client Logs and Other Files

- Installation log file: `/system/volatile/install_log`
- AI manifest downloaded (server):
`/system/volatile/ai.xml`
- Derived manifest (if used):
`/system/volatile/manifest.xml`
- Configuration profiles downloaded from server:
`/system/volatile/profile/*`
- List of AI services: `/system/volatile/service_list`
- SMF service log for manifest/profile locator:
`var/svc/log/application-manifest-locator:default.log`
- SMF service log for AI application:
`/var/svc/log/application-auto-installer:default.log`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When an installation fails, one of the first actions you should perform is to check the logs and error messages. This slide contains a list of all the files that are on the AI client that you can use to help you troubleshoot a client installation failure.

Gathering AI Server Logs and Other Files

- AI Apache webserver log files:
/var/ai/image-server/logs/access_log
/var/ai/image-server/logs/error_log
- AI install server SMF service log
/var/svc/log/system-install-server:default.log
- Boot configuration files
/etc/netboot/

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide contains a list of all the files that are on the AI client that you can use to help you troubleshoot AI server issues.

Using Additional Debugging Tools

- For `installadm` debug information, review the log files:
 - `/system/volatile` (during installation)
 - `/var/log/install` (after installation)
- To turn on additional debug for SPARC boot environment operations, use the `install_debug` boot argument:
`ok> boot net:dhcp - install install_debug`
- For disk-related failures, gather additional data by using:
`/usr/lib/python2.6/vendor-packages/solaris_install/target/discovery.py`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For troubleshooting, AI always has debug information in the log files, but you can turn on additional information by using the options listed in the slide.

Starting with the Oracle Solaris 11.1 release, the log files for `installadm` are now in `/system/volatile` during installation and in `/var/log/install` after installation.

Note for second bullet: To turn on additional debug for the X86 boot environment operations, add the following to the kernel line boot entry: `install_debug=enable`.

For disk-related failures, you can use the command presented in the third bullet to gather data about disk issues.

Checking DNS

Check that the DNS server is properly configured:

- A non-empty `resolv.conf` file exists and is properly configured.
- If the `resolv.conf` file does not exist or is empty, run the following command:

```
# /sbin/dhccpinfo DNSserv
```
- If an `/etc/resolv.conf` file exists and is properly configured, check for the following problems:
 - The DNS server cannot resolve the IPS repository server name.
 - There is no default route to reach the DNS server.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

An incorrectly configured DNS server can also contribute to a failed installation. To check whether DNS is configured on your client correctly, verify that a non-empty `/etc/resolv.conf` file exists.

Note for second bullet: This command checks that your DHCP server is providing DNS server information to the client. If the command returns nothing, the DHCP server is not set up to provide DNS server information to the client. Contact your DHCP administrator to correct this problem.

SPARC Network Booting Errors and Possible Causes

- Timed out waiting for BOOTP/DHCP reply
 - Probable cause: DHCP configuration problem
 - Corrective action: Check whether your client is configured correctly in the DHCP server.
- Boot load failed
 - Probable cause: Client DHCP configuration problem or another DHCP server is responding to the client
 - Corrective action: Check the DHCP configuration for this client, or determine the presence of another DHCP server in the subnet.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide presents a number of errors or problems you might see when booting a SPARC client over the network and their possible causes.

- **Timed out waiting for BOOTP/DHCP reply:** If a DHCP server is not responding to a SPARC client's request, the a timeout message is displayed. This message indicates that the client is sending a DHCP request and no response has been made to that request. This error is probably caused by a DHCP configuration problem. Check whether your client is configured correctly in the DHCP server.
- **Boot load failed:** If the AI client starts downloading the `boot_archive`, but then fails with the error, "Boot load failed," that indicates that the client DHCP information is configured incorrectly. This error could happen if another DHCP server is responding to the client. Check the DHCP configuration for this client. If the configuration appears to be correct, determine whether another DHCP server is in the subnet.

SPARC Network Booting Errors and Possible Causes

- Internal server error or WAN boot alert
 - Probable cause: AI client cannot find the `boot_archive` or cannot access the file.
 - Corrective action: Fix the client's `boot_archive` file
- Error Message 403: Forbidden or 404 Not Found
 - Probable cause:
 - Incorrect WAN boot image path configuration
 - Non-existent or incomplete image path
 - No access due to permission issues
 - Corrective action: Check the following:
 - DHCP configuration
 - Contents of the net image you specified when you ran `installadm create-service`
 - WAN boot configuration

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- **Internal server error or WAN boot alert:** After the AI client has obtained the IP address and initial parameters to start downloading the boot archive, the client might be unable to find or access the `boot_archive`. For both of these problems, you need to fix the `boot_archive` file configured for this client by checking the path name and permissions of the `boot_archive` at `$IMAGE/boot/boot_archive`.
- **Error Message 403: Forbidden or 404 Not Found:** The messages ERROR 403: Forbidden and ERROR 404: Not Found are displayed if the AI client successfully downloads the `boot_archive` and boots the Oracle Solaris kernel but fails to get one of the image archives. An error message is displayed indicating which file is causing the problem. One of the following conditions could be causing the problem:
 - The image path configured in WAN boot is not correct.
 - The image path does not exist or is incomplete.
 - Access is denied due to permission issues.

Check your DHCP configuration or the contents of the net image you specified when you ran `installadm create-service`. Check your WAN boot configuration.

SPARC Error Messages

- Automated Installation Failed Message
 - Cause: AI installation failed
 - Corrective action: Refer to the `/system/volatile/install_log`.
- Unable To Contact Valid Package Server
 - Cause: Client cannot access the package repository
 - Corrective action: Check the `/system/volatile/install_log` file for `TransportFailures` error messages.
- Package Not Found
 - Cause: A package specified in the AI manifest cannot be located in the IPS repositories
 - Corrective action: Verify the validity of the package, and if appropriate, add IPS repository in the AI manifest.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note: The errors listed in the slide are common to both SPARC and x86 installations.

Notes for Unable To Contact Valid Package Server error message: The installation client needs to reach the IPS package repository defined in the AI manifest in order to install the Oracle Solaris OS. If the client cannot access the package repository, the installation fails and the `application/auto-installer` service transitions to maintenance. Check the `/system/volatile/install_log` file. Depending on which messages you see, do the following:

- Try to reach the package server from the failed client system, for example, by using `ping`.
- If you are using DNS, check whether DNS is correctly configured on the AI client.
- If you are using a local repository, check whether you have made the repository accessible to all clients.
- Make sure the URI in the AI manifest does not have a typographical error.
- Use a command such as the following command to check whether the package repository is valid:

```
$ pkg list -g http://pkg.example.com/solaris/ entire
```

You might need to refresh the catalog or rebuild the index.

Notes for Package Not Found error message: If one of the packages specified in the AI manifest cannot be located in the IPS repositories, then the installer fails before installing any packages on the disk. Check whether the package in question is a valid package. If this package is available from a different IPS repository, add that IPS repository in the AI manifest by adding another publisher element to the source element.

AI Client Boot Failure

- SPARC-based system boot failure

```
OpenBoot 4.23.4, 8184 MB memory available, Serial #69329298.  
Ethernet address 0:14:4f:21:e1:92, Host ID: 8421e192.  
Rebooting with command: boot net:dhcp - install  
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:  
1000 Mbps FDX Link up  
Timed out waiting for BOOTP/DHCP reply  
Timed out waiting for BOOTP/DHCP reply
```

- x86-based system boot failure

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)  
Copyright(C) 1997-2007, Intel Corporation  
CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF  
FFFF 7BDA264F1400  
DHCP..... No DHCP or ProxyDHCP offers were received PXE-  
MOF: Exiting Intel Boot Agent
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If a DHCP server is not responding to an AI client's request, the messages in this example are displayed.

The timeout message indicates that the client is sending a DHCP request, and no response has been made to that request. This error is probably caused by a DHCP configuration problem. Check whether your client is configured correctly in the DHCP server.

Verifying the Local DHCP Configuration

```
# installadm create-service -n x86_ai \
-s /var/tmp/sol-11-1111-ai-x86.iso \
-i 192.168.0.130 -c 10 -d /export/ai/x86_ai
...
# installadm create-client -e 08:00:27:85:C7:D6 -n x86_ai
...
# tail -15 /etc/inet/dhcd4.conf
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.130 192.168.0.139;
    option broadcast-address 192.168.0.255;
    option routers 0.0.0.0;
    next-server 192.168.0.100;
}
class "PXEBoot" {
    match if (substring(option vendor-class-identifier, 0, 9) =
"PXEClient");
filename "default-i386/boot/grub/pxegrub";
}
host 08002785C7D6 {
    hardware ethernet 08:00:27:85:C7:D6;
    filename "0108002785C7D6";
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By default, AI uses the Internet Systems Consortium (ISC) file `dhcd4.conf` to configure DHCP services. In this example, the AI service `x86_ai` is configured with a DHCP IP address range of 192.168.0.130 through 192.168.0.139 (installadm command options: `-i 192.168.0.130 -c 10`).

Next, an AI client with an Ethernet address of 08:00:27:85:C7:D6 has been added to the AI service. Using the `tail` command, you can verify that the correct IP address range has been added to the `dhcd4.conf` file.

Deploying the OS Using AI

```

10:12:02 0% Preparing for Installation
10:12:02 0% Preparing for Installation
10:12:02 0% Preparing for Installation
10:12:02 1% Preparing for Installation)10:12:02 1% Preparing for Installat
ion)10:12:02 1% Preparing for Installation)10:12:02 2% Preparing for Insta
llation)10:12:02 2% Preparing for Installation)10:12:02 2% Preparing for I
nstallation)10:12:03 3% Preparing for Installation
10:12:03 3% Preparing for Installation
10:12:03 3% Preparing for Installation
10:12:03 4% Preparing for Installation
10:12:03 4% Preparing for Installation)10:12:03 4% Preparing for Installat
ion
10:12:10 7% target-discovery completed.
10:12:10 == Executing Target Selection Checkpoint ==
10:12:10 Selected Disk(s) : c3t0d0
10:12:10 13% target-selection completed.
10:12:11 17% ai-configuration completed.
10:12:11 19% var-shared-dataset completed.
10:12:21 21% target-instantiation completed.
10:12:21 21% Beginning IPS transfer
10:12:22 Creating IPS image
10:12:25 Installing packages from:
10:12:25 solaris
10:12:25 origin: http://s11-serv1.mydomain.com/

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After the AI service has been configured for nonglobal zone deployment, net-boot the install client.

The screenshot in this example is taken from the AI server. It shows a critical event during automatic installation from the server. The circle shows that AI has successfully contacted the IPS server and is currently installing software packages.

Deploying Non-Global Zones Using AI

```
14:42:02 1% Preparing for Installation
14:42:02 2% Preparing for Installation
14:42:02 2% Preparing for Installation
14:42:02 2% Preparing for Installation
14:42:02 3% Preparing for Installation
14:42:02 3% Preparing for Installation
14:42:02 3% Preparing for Installation
14:42:02 4% Preparing for Installation
14:42:02 4% Preparing for Installation
14:42:02 4% Preparing for Installation
14:42:09 7% target-discovery completed.
14:42:09 == Executing Target Selection Checkpoint ==
14:42:09 Selected Disk(s) : c3t0d0
14:42:09 13% target-selection completed.
14:42:09 Zone name: engineering
14:42:09 source: http://web.mydomain.com/zone_configs/engineering.cfg
14:42:11 17% ai-configuration completed.
14:42:11 19% var-shared-dataset completed.
14:42:20 21% target-instantiation completed.
14:42:20 21% Beginning IPS transfer
14:42:20 Creating IPS image
14:42:22 Installing packages from:
14:42:22 solaris
14:42:22 origin: http://s11-ss.mydomain.com/
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After the AI service has been configured for nonglobal zone deployment, net-boot the install client.

The screenshot in this example is taken from the AI server. It shows two critical events during the automatic installation. In the top red circle, AI has obtained a valid zone configuration file for the web.mydomain.com server. The bottom circle shows that AI has successfully contacted the IPS server and is currently installing software packages.

After AI has completed the installation, the target system must be rebooted. Upon rebooting, the nonglobal zone installation begins. Note that the nonglobal zone installation will take several minutes to complete.

Common AI Problems

- An invalid manifest (including zones) aborts the automated installation.
- Invalid client criteria.
- An invalid client SC profile.
- Common causes for the failure:
 - Typos
 - Missing elements
 - Incorrect syntax
 - Improper XML code
 - Moved or removed file references
- Use the various message logs associated with AI technology.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The most common areas in which errors can be introduced are the AI manifest, client criteria, and SC profile.

Common causes are typographical errors (typos), missing elements, incorrect syntax, and improper use of XML code. Moved or removed files that are references can also cause problems.

To investigate the cause of the validation failure, use the various message logs associated with AI technology.

After you have an idea of what is caused the failure, you can correct the errors.

Agenda

- Reviewing AI Features and Functionality
- Troubleshooting Client Installation Failures
- Case Study 5

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: AI Fails to Deploy Solaris OS!

- Clarify problem statement.
 - System `host2` fails to deploy the Solaris 11 OS
- Using “assumptive” method, prioritize the most probable causes of failure.
 - AI service configuration
 - DHCP fault
 - IPS fault
 - Hardware fault
- Starting with the most probable cause of failure, gather pertinent information.
 - Replicate the error.
 - Peruse AI logs.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Troubleshooting Steps

Perform the following steps to identify the cause of the problem:

1. Verify that the AI client fails to install the OS.
2. Search the AI log for error messages (if available).
3. Check DHCP status (if appropriate).
4. Check IPS status (if appropriate).
5. Verify AI configuration (if appropriate).
6. Repair the fault.
7. Verify the solution.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the high-level steps involved in this case study.

Case Study 5: Walkthrough

```
##### Verify that the AI client fails the OS installation. #####
{0} ok banner

SPARC Enterprise T5120, No Keyboard
Copyright (c) 1998, 2012, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.33.6.b, 1920 MB memory available, Serial #83496672.
Ethernet address 0:14:4f:fa:e:e0, Host ID: 84fa0ee0.

{0} ok boot net:dhcp - install
Boot device: /virtual-devices@100/channel-devices@200/network@0:dhcp File and
args: - install
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: configuring /virtual-devices@100/channel-
devices@200/network@0:dhcp

<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 368 of 368 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Wed Mar 27 22:33:25 wanboot progress: miniroot: Read 245893 of 245893 kB (100%)
Wed Mar 27 22:33:25 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide begin the Case Study 4 walkthrough. Note that line beginning with “#####” are comments.

Case Study 5: Walkthrough

```

Downloading solaris.zlib
--2013-06-12 15:57:45-- http://192.168.124.21:5555/export/ai/solaris11_1_6_2_0-
sparc//solaris.zlib
Connecting to 192.168.124.21:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 132515328 (126M) [text/plain]
Saving to: `/tmp/solaris.zlib'
100%[=====>] 132,515,328 11.2M/s in 11s
2013-06-12 15:57:57 (11.1 MB/s) - `/tmp/solaris.zlib' saved [132515328/132515328]
Downloading solarismisc.zlib
--2013-06-12 15:57:57-- http://192.168.124.21:5555/export/ai/solaris11_1_6_2_0-
sparc//solarismisc.zlib
Connecting to 192.168.124.21:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12403712 (12M) [text/plain]
Saving to: `/tmp/solarismisc.zlib'
100%[=====>] 12,403,712 11.1M/s in 1.1s
2013-06-12 15:57:58 (11.1 MB/s) - `/tmp/solarismisc.zlib' saved [12403712/12403712]
Downloading .image_info
--2013-06-12 15:57:58-- http://192.168.124.21:5555/export/ai/solaris11_1_6_2_0-
sparc//.image_info
Connecting to 192.168.124.21:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 87 [text/plain]
Saving to: `/tmp/.image_info'
100%[=====>] 87 --.-K/s in 0s
2013-06-12 15:57:58 (2.23 MB/s) - `/tmp/.image_info' saved [87/87]

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
Done mounting image
Configuring devices.
Hostname: solaris
Service discovery phase initiated
Service name to look up: default-sparc
Service discovery finished successfully
Process of obtaining install manifest initiated
Using the install manifest obtained via service discovery
Automated Installation started
The progress of the Automated Installation will be output to the console
Detailed logging is in the logfile at /system/volatile/install_log
Press RETURN to get a login prompt at any time.
```

```
solaris console login:
15:59:11 Using XML Manifest: /system/volatile/ai.xml
15:59:11 Using profile specification: /system/volatile/profile
15:59:11 Using service list file: /var/run/service_list
15:59:11 Starting installation.
15:59:11 0% Preparing for Installation
15:59:11 100% manifest-parser completed.
15:59:11 0% Preparing for Installation
15:59:12 1% Preparing for Installation
15:59:12 2% Preparing for Installation
15:59:12 3% Preparing for Installation
15:59:12 4% Preparing for Installation
15:59:13 7% target-discovery completed.
15:59:13 Selected Disk(s) : c3d0
15:59:13 13% target-selection completed.
15:59:14 17% ai-configuration completed.
15:59:14 19% var-share-dataset completed.
15:59:29 21% target-instantiation completed.
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to its upper right. The logo is white and is positioned on a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
15:59:29    21% Beginning IPS transfer
15:59:29    Creating IPS image
##### The OS installation fails here. #####
16:00:00    Error occurred during execution of 'generated-transfer-1165-1' checkpoint.
16:00:01    Failed Checkpoints:
16:00:01        generated-transfer-1165-1
16:00:01    Checkpoint execution error:
16:00:01        Framework error: code: 6 reason: Couldn't resolve host 'pkg.oracle.com'
16:00:01        URL: 'http://pkg.oracle.com/solaris/release/versions/0/'
16:00:01    Automated Installation Failed. See install log at
            /system/volatile//install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
Jun 12 16:00:01 solaris svc.startd[9]: application/auto-installer:default failed fatally:
            transitioned to maintenance (see 'svcs -xv' for details)

SUNW-MSG-ID: SMF-8000-YX, TYPE: defect, VER: 1, SEVERITY: major
EVENT-TIME: Wed Jun 12 16:00:01 UTC 2013
PLATFORM: SPARC-Enterprise-T5120, CSN: unknown, HOSTNAME: solaris
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: bd0e83b3-f365-c10e-9e42-ef08c926e9bc
DESC: A service failed - a start, stop or refresh method failed.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/application/auto-installer:default is unavailable.
REC-ACTION: Run 'svcs -xv svc:/application/auto-installer:default' 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.
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Log in to the AI client and the auto-installer service status. #####
solaris console login: root
Password: solaris
'Jun 12 16:00:32 solaris login: ROOT LOGIN /dev/console
Oracle Corporation      SunOS 5.11      11.1      March 2013
root@solaris:~# svcs -xv svc:/application/auto-installer:default
svc:/application/auto-installer:default (automated installer)
  State: maintenance since Wed Jun 12 16:00:01 2013
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
  See: http://support.oracle.com/msg/SMF-8000-KS
  See: man -M /usr/share/man -s 1M auto-install
  See: /var/svc/log/application-auto-installer:default.log
Impact: This service is not running.
##### Check the AI log for error indications. #####
root@solaris:~# cat /var/svc/log/application-auto-installer:default.log
[ Jun 12 15:58:57 Executing start method ("/lib/svc/method/auto-installer"). ]
...
15:59:29      Creating IPS image
16:00:00      Error occurred during execution of 'generated-transfer-1165-1' checkpoint.
16:00:01      Failed Checkpoints:
16:00:01          generated-transfer-1165-1
16:00:01      Checkpoint execution error:
16:00:01          Framework error: code: 6 reason: Couldn't resolve host 'pkg.oracle.com'
16:00:01          URL: 'http://pkg.oracle.com/solaris/release/versions/0/'
16:00:01      Automated Installation Failed.  See install log at
              /system/volatile//install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
[ Jun 12 16:00:01 Method "start" exited with status 95. ]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Based on the error indication, check the IPS functionality on the AI client. #####
root@solaris:~# pkg publisher solaris

    Publisher: solaris
      Alias:
Origin URI: http://pkg.oracle.com/solaris/release/
  SSL Key: None
  SSL Cert: None
Client UUID: 84c7a57e-90cf-11e2-a91e-80144f3be152
Catalog Updated: Wed Jun 12 16:48:43 2013
  Enabled: Yes
root@solaris:~# pkg search entire
pkg: Some repositories failed to respond appropriately:
solaris:
Unable to contact valid package repository
Encountered the following error(s):
Unable to contact any configured publishers.
This is likely a network configuration problem.
Framework error: code: 6 reason: Couldn't resolve host 'pkg.oracle.com'
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right of the "E".

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Check the IPS functionality on the AI server. #####
root@host02:~# pkg publisher solaris

    Publisher: solaris
    Alias:
    Origin URI: file:///net/192.168.124.1/seif/os/s11_ul-sru7.2_repo/
    SSL Key: None
    SSL Cert: None
    Client UUID: 85f2f2ac-90fa-11e2-871d-00144f025cf7
    Catalog Updated: April 18, 2013 02:12:09 AM
    Enabled: Yes

root@host02:~# pkg search entire
INDEX          ACTION VALUE
    PACKAGE
pkg.description set    Provides for power management support of the entire operating
                        system, including the configuration of the maximum time allowed to reach both minimum
                        and full capacity, and whether or not to permit system suspend and resume if the
                        platform supports it.
pkg:/system/kernel/power@0.5.11-0.175.1.0.0.24.2
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Check the AI service configuration. #####
root@host02:~# installadm list -cmp
```

Service Name	Client Address	Arch	Image Path
default-sparc	00:14:4F:FA:0E:E0	sparc	/export/ai/solaris11_1_6_2_0-sparc

Service/Manifest Name	Status	Criteria
default-sparc		
sparc-manifest		platform = SUNWSPARC-SUNFIRE-T5120
orig_default	Default	None
solaris11_1_6_2_0-sparc		
orig_default	Default	None

Service/Profile Name	Criteria
default-sparc	
ldom1-profile	mac = 00:14:4F:FA:0E:E



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Check the contents of the AI client user-defined manifest. #####
root@host02:~# installadm export -n default-sparc -m sparc-manifest
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="sparc-manifest">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <!--
            Subsequent <filesystem> entries instruct an installer to create
            following ZFS datasets:

                <root_pool>/export          (mounted on /export)
                <root_pool>/export/home      (mounted on /export/home)
          ...

        <source>
          <publisher name="solaris">
            <origin name="file:///net/192.168.124.1/seif/os/s11_u1-sru7.2_repo"/>
          </publisher>
        </source>
      </logical>
    </target>
  </ai_instance>
##### Note that the manifest IPS origin name is different from the one sent #####
##### to the AI client. #####
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Check the contents of the default manifest. #####
root@host02:~# cd /export/ai/solaris11_1_6_2_0-sparc/auto_install/manifest
root@host02:/export/ai/solaris11_1_6_2_0-sparc/auto_install/manifest# ls
ai_manifest.xml    default.xml        zone_default.xml

root@host02:/export/ai/solaris11_1_6_2_0-sparc/auto_install/manifest# cat default.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--

  Copyright (c) 2008, 2012, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
...
          <source>
            <publisher name="solaris">
              <origin name="http://pkg.oracle.com/solaris/release"/>
            </publisher>
          </source>
##### Note that the default manifest IPS origin name matches the one sent #####
##### to the AI client. This indicates that the AI client failed to meet the #####
##### criteria for the user-defined manifest.
...

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Verify the criteria for the user-defined manifest. #####
root@host02:~# installadm list -m
Service/Manifest Name      Status      Criteria
-----
default-sparc
  sparc-manifest           platform = SUNWSPARC-SUNFIRE-T5120
  orig_default             Default    None
...
root@solaris:~# uname -i
SUNW,SPARC-Enterprise-T5120
##### Note that the manifest criteria (platform) in the AI service does #####
##### not match the AI client. Fix the problem. #####
root@host02:~# installadm export -n default-sparc -m sparc-manifest > \
/var/tmp/manifest_tmp1.xml
root@host02:~# installadm delete-manifest -m sparc-manifest -n default-sparc
root@host02:~# installadm create-manifest -m sparc-manifest \
-f /var/tmp/manifest_tmp1.xml
-c platform="SUNW,SPARC-Enterprise-T5120" \
-n default-sparc
root@host02:~# installadm list -m
Service/Manifest Name      Status      Criteria
-----
default-sparc
  sparc-manifest           platform = SUNWSPARC-Enterprise-T5120
  orig_default             Default    None

solaris11_1_6_2_0-sparc
  orig_default             Default    None
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 5: Walkthrough

```
##### Verify that your solution solves the problem. #####
{0} ok boot net:dhcp - install
Boot device: /virtual-devices@100/channel-devices@200/network@0:dhcp  File and args: -
install
...
16:27:09      21% Beginning IPS transfer
16:27:09      Creating IPS image
16:27:20      Startup: Retrieving catalog 'solaris' ... Done
16:27:25      Startup: Caching catalogs ... Done
16:27:26      Startup: Refreshing catalog 'solaris' ... Done
16:27:26      Installing packages from:
16:27:26          solaris
16:27:26              origin:  file:///net/192.168.124.1/seif/os/s11_u1-sru7.2_repo/
16:27:27      Startup: Refreshing catalog 'solaris' ... Done
...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif, uppercase font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

More Information

- Oracle's Bugster Categories/Subcategories:
 - Automated Install (Client): install/autoinstall
 - Automated Install (Server): install/installadm
 - First boot: solaris/install/configuration
 - Target discovery/instantiation: solaris/install/targets
- Manual pages:
 - installadm(1M)
 - ai_manifest(4) – documents the XML manifest
 - aimanifest(1M) – documents the derived manifest
 - sysconfig(1M)

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Bugster categories and subcategories for AI and the related man pages are listed in the slide.

Additional Resources

- Install Cheat Sheet [Note: Includes troubleshooting tips]:
<http://www.oracle.com/technetwork/server-storage/solaris11/documentation/solaris-11-install-heat-sheet-1609420.pdf>
- OTN AI Page (white papers, blogs, etc.):
<http://www.oracle.com/technetwork/server-storage/solaris11/technologies/modernizedinstaller-461041.html>
- Documentation: Installing Oracle Solaris 11.1 Systems:
http://docs.oracle.com/cd/E26502_01/html/E28980/index.html

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

During an AI client installation, you receive an internal server error or WAN boot alert. What are the probable causes of this problem? (Select two.)

- a. AI client cannot find the `boot_archive` file.
- b. The client DHCP configuration is incorrect.
- c. There is a permissions issue.
- d. AI client cannot access the `boot_archive` file.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a,d

Quiz

During an AI installation, you receive an Automated Installation Failed Message. What action should you take first to start troubleshooting the problem?

- a. Check the AI manifest for errors.
- b. Refer to the `/system/volatile/install_log`.
- c. Review the `installadm(1M)` man page.
- d. Try running the installation again but this time with the `install_debug` boot argument.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Practice 6: Overview

- 6-1: Troubleshooting a failed AI installation fails to complete
- 6-2: Troubleshooting a failed zone deployment during an AI installation

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Recall the Automated Installer (AI) features and functions
- Troubleshoot client installation failures

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

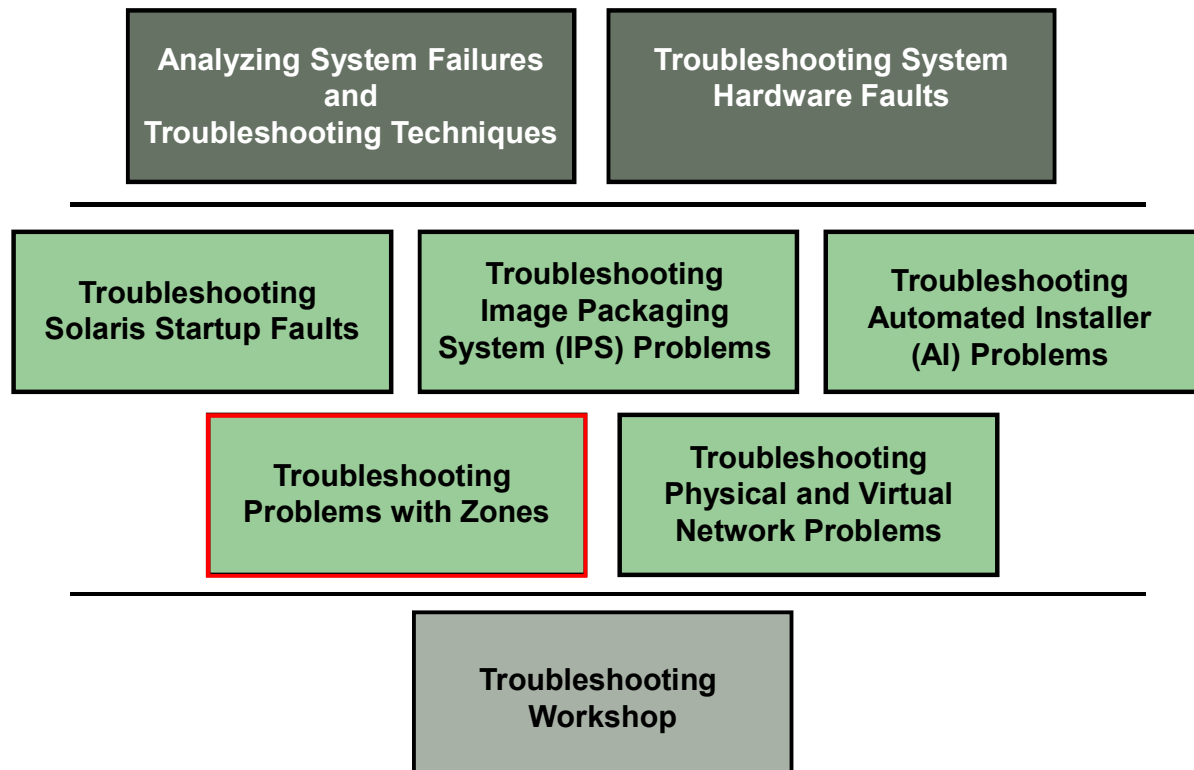
Troubleshooting Problems with Zones



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this lesson, Troubleshooting Problems with Zones, we focus on potential zones issues, their causes, and how to troubleshoot and resolve these issues.

Objectives

After completing this lesson, you should be able to:

- Recall the zones features and functionality
- Verify the zone configuration
- Shut down a non-global zone
- Halt a non-global zone
- Uninstall a non-global zone
- Clone a non-global zone
- Troubleshoot a zone that does not halt
- Identify zones-related SMF services
- Use SMF troubleshooting techniques within a zone
- Monitor zones
- Fix IPS problems with non-global zones

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Reviewing Zones Features and Functionality
- Checking the Basics
- Using Zones-related SMF Services
- Monitoring Non-Global Zones
- Fixing IPS Problems with Non-Global Zones
- Case Study 6

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Global and Non-Global Zone Features and Primary Functions

- Every Oracle Solaris system contains a global zone.
 - Default zone for the system
 - Used for system-wide administrative control
 - Only zone that is bootable from the system hardware
- A non-global zone isolates software applications or services by using flexible, software-defined boundaries.
- A process running in a non-global zone:
 - Can manipulate, monitor, and directly communicate with other processes that are assigned to the same zone.
 - Cannot perform these functions with processes that are assigned to other zones.
- Each zone is assigned a zone name and unique numeric identifier.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Every Oracle Solaris system contains a global zone. The global zone is both the default zone for the system and the zone used for system-wide administrative control. Only the global zone is bootable from the system hardware.

A non-global zone functions as a container. One or more applications can run in this container without interacting with the rest of the system. Zones isolate software applications or services by using flexible, software-defined boundaries. Applications that are running in the same instance of the Oracle Solaris operating system can then be managed independently of one other. That is, different versions of the same application can be run in different zones to match the requirements of your configuration.

A process assigned to a zone can manipulate, monitor, and directly communicate with other processes that are assigned to the same zone. However, the process cannot perform these functions with processes that are assigned to other zones in the system or with processes that are not assigned to a zone. Processes that are assigned to different zones are only able to communicate through network APIs.

Each zone, including the global zone, is assigned a zone name. The global zone always has the name `global`. Each zone is also given a unique numeric identifier, which is assigned by the system when the zone is booted. The global zone is always mapped to ID 0.

Reviewing Immutable Zones Technology

- By default, a zone is configured to have a writable root dataset.
- Using the `zonecfg` utility and the `file-mac-profile` property, a zone can be configured with one of the options presented in the table below.

	<code>/, /usr, /lib, ...</code>	<code>/etc</code>	<code>/var</code>	other
None	Writeable	Writeable	Writeable	Writeable
Flexible	Read-only	Writeable	Writeable	Read-only
Fixed	Read-only	Read-only	Writeable	Read-only
Strict	Read-only	Read-only	Read-only	Read-only

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

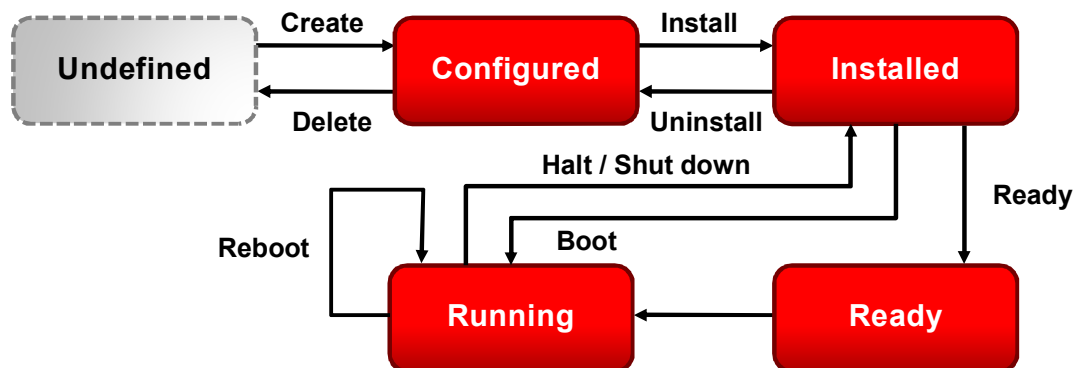
Introduced in Oracle Solaris 11, the immutable zones technology enables system administrators to protect a non-global zone's configuration by implementing read-only file system profiles. Unless performed as specific maintenance operations, modifications to system binaries or system configurations within these zones are blocked.

By default, a zone is configured to have a writable root dataset. Using the `zonecfg` utility and the `file-mac-profile` property, system administrators can configure a zone with one of the following options:

- **none:** A standard, read-write, non-global zone, with no additional protection beyond the existing zones boundaries
- **flexible-configuration:** Permits modification of files in `/etc/*` directories, changes to root's home directory, and updates to `/var/*` directories
- **fixed-configuration:** Permits updates to `/var/*` directories, with the exception of directories that contain system configuration components. IPS packages, including new packages, cannot be installed. Persistently enabled SMF services are fixed. SMF manifests cannot be added from the default locations. Logging and auditing configuration files can be local. syslog and audit configuration are fixed.

- **strict:** A read-only file system with no exceptions. IPS packages cannot be installed. Persistently enabled SMF services are fixed. SMF manifests cannot be added from the default locations. Logging and auditing configuration files are fixed. Data can only be logged remotely.

Reviewing Non-Global Zone States



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

As you configure, enable, and use a non-global zone, its status changes. The possible non-global zone states are as follows:

- **Undefined:** In this state, the zone's configuration has not been completed and committed to stable storage. This state also occurs when a zone's configuration has been deleted.
- **Configured:** In this state, the zone's configuration is complete and committed to stable storage. However, those elements of the zone's application environment that must be specified after initial boot are not yet present.
- **Incomplete:** This is a transitional state. During an install or uninstall operation, the state of the target zone is set to incomplete. After successful completion of the operation, the state is set to the correct state. However, a zone that is unable to complete the install process will stop in this state.
- **Installed:** In this state, the zone configuration is instantiated on the system. At this point, the system administrator verifies that the configuration can be successfully used on the designated Oracle Solaris system. Packages are installed under the zone's root path. In this state, the zone has no associated virtual platform.

- **Ready:** In this state, the virtual platform for the zone is established. The kernel creates the zone scheduling process, network interfaces are set up and made available to the zone, file systems are mounted, and devices are configured. A unique zone ID is assigned by the system. At this stage, no processes associated with the zone have been started.
- **Running:** In this state, the user processes associated with the zone application environment are running. The zone enters the running state as soon as the first user process associated with the application environment (`init`) is created.
- **Shutting down, down:** These states are transitional states that are visible while the zone is being halted. However, a zone that is unable to shut down for any reason will stop in one of these states.

Reviewing How Zones Work: Global Zone

Global Zone Functions
Provides the single instance of the Oracle Solaris kernel that is bootable and running on the system
Contains a complete installation of the Oracle Solaris system software packages
Can contain additional software packages or additional software, directories, files, and other data not installed through packages
Provides a complete and consistent product database that contains information about all software components installed in the global zone
Holds configuration information specific to the global zone only, such as the global zone host name and file system table
Is the only zone that is aware of all devices and all file systems
Is the only zone with knowledge of non-global zone existence and configuration
Is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The table in this slide and the next highlight the functional relationship and differences between the global zone and non-global zones.

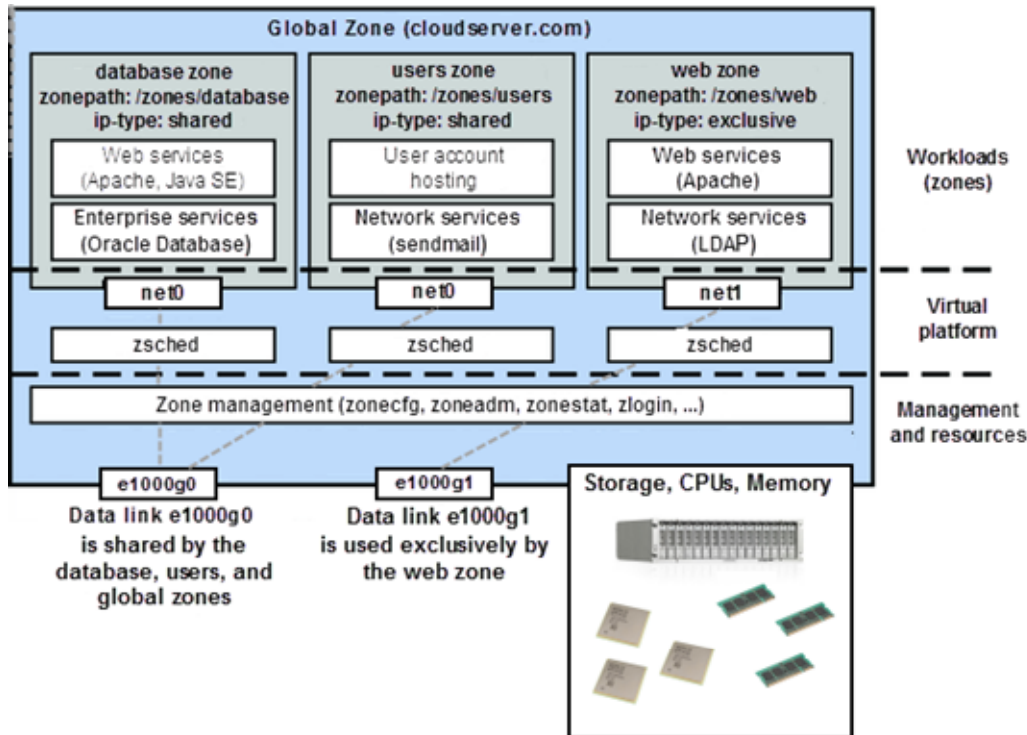
Reviewing How Zones Work: Non-Global

Non-Global Zone Functions
Shares operation under the Oracle Solaris kernel booted from the global zone
Contains an installed subset of the complete Oracle Solaris operating system software packages
Can contain additional installed software packages
Can contain additional software, directories, files, and other data created on the non-global zone that are not installed through packages
Has a complete and consistent product database that contains information about all software components installed on the zone
Is not aware of the existence of any other zones
Cannot install, manage, or uninstall other zones, including itself
Has configuration information specific to that non-global zone only, such as the non-global zone host name and file system table
Is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right of the "E".

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Reviewing How Zones Work: Example



ORACLE

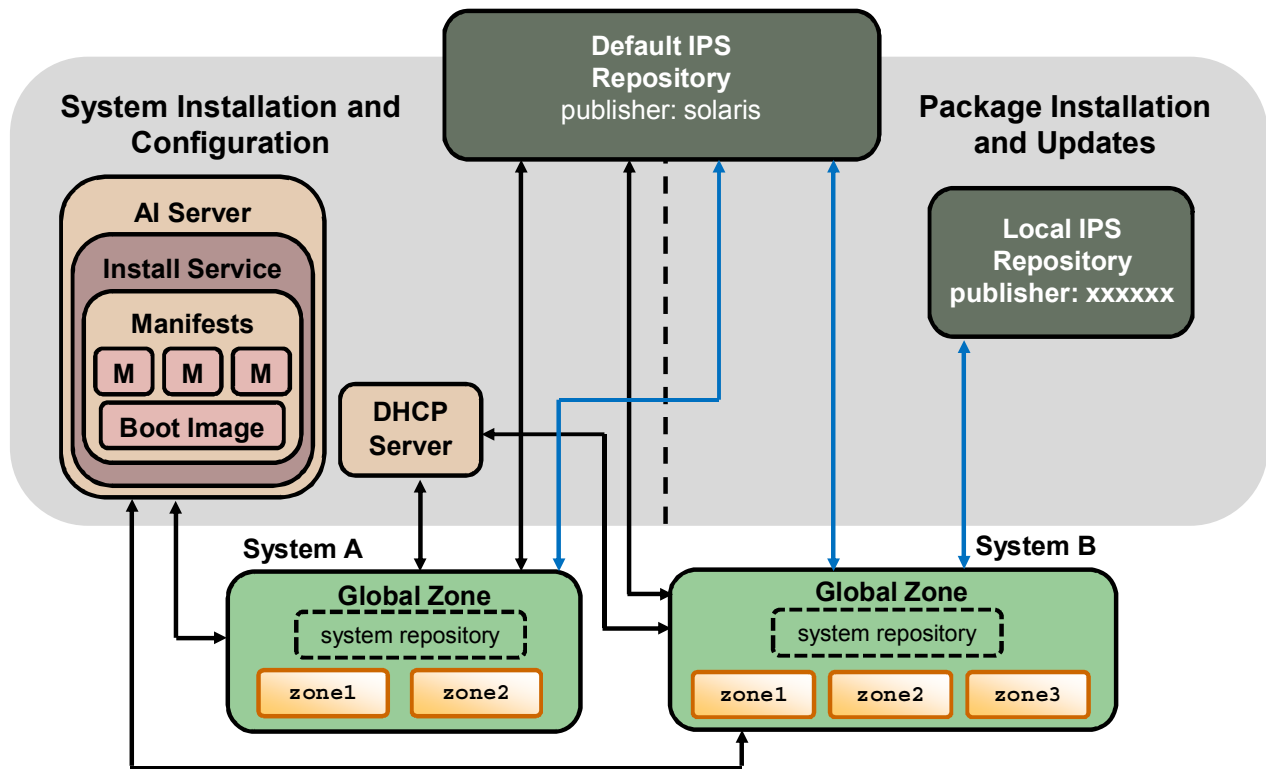
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Each zone, including the global zone, is assigned a zone name. The global zone always has the name global. Each zone is also given a unique numeric identifier, which is assigned by the system when the zone is booted. The global zone is always mapped to ID 0.

The figure in this slide shows a system with three zones (database, users, and web). Each of the zones is running a workload unrelated to the workloads of the other zones. This example illustrates that different application can be run without negative consequences in different zones, to match the consolidation requirements. Each zone can provide a customized set of services.

Each zone workload can be allocated system resources such as networks, storage, CPUs, and memory to meet performance requirements.

Reviewing How Zones Work with AI, IPS, and SMF



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Over the next few slides, we are going to review how IPS, AI, and SMF work together to install and configure zones. We'll then review how IPS, SMF, and zones work together to install and update packages.

Reviewing How AI Installs Non-Global Zones

1. When a system is installed using AI, non-global zones can be installed on that system by using the `configuration` element in the AI manifest.
2. When the system first boots after the global zone installation, the zone's self-assembly SMF service (`svc:/system/zones-install:default`) configures and installs each non-global zone defined in the global zone AI manifest.
3. If the zone is configured with `autoboot=true`, the `system/zones-install` service boots the zone after the zone is installed.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note for step 1: If you do not provide a custom AI manifest for a non-global zone, the default AI manifest for zones is used for the installation:

`/usr/share/auto_install/manifest/zone_default.xml`.

Reviewing How to Specify Non-Global Zones in the Global Zone AI Manifest

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance>
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@latest</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
    <configuration type="zone" name="zone1" source="http://server/zone1/config"/>
    <configuration type="zone" name="zone2" source="file:///net/server/zone2/config"/>
  </ai_instance>
</auto_install>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A sample AI manifest is presented in the slide. Note how the `configuration` element in the AI manifest for the client system is used to specify the non-global zones. The `name` attribute is used to specify the name of the zone while the `source` attribute is used to specify the location of the `config` file for the zone. The source location can be any `http://` or `file://` location that the client can access during installation.

Troubleshooting Non-Global Zones Installation Errors

- Monitor non-global zone installation by:
 - Monitoring the `system/zones-install` service
 - Checking the output of `zoneadm list -cv`
- Errors that result in zone installation failures:
 - A zone `config` file is not syntactically correct.
 - A collision exists among zone names, zone paths, or delegated ZFS datasets in the set of zones to be installed.
 - Required zone resources are not configured in the global zone.
 - The zone fails to install using AI.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can monitor non-global zone installation by monitoring the `system/zones-install` service or by checking the output of `zoneadm list -cv`.

If the non-global zone installation fails, check for the following errors:

- A zone `config` file is not syntactically correct. To correct this error, be sure to use the `zonecfg -z zonename verify` command to check the syntax.
- A collision exists among zone names, zone paths, or delegated ZFS datasets in the set of zones to be installed. To correct this error, run the `zonecfg -z zonename info` command for each zone in the system to identify the zone names, paths, and datasets. Verify that the collision exists. Correct the discrepancy as necessary.
- Required zone resources, such as datasets, are not correctly configured in the global zone. To correct this error, run the `zonecfg -z zonename info` command to list the zone resource configuration. Verify that resources stated in the zone configuration match the existing resources in the global zone. Correct the discrepancy as necessary.
- The zone fails to install using AI. If the problem occurs during AI installation of the zone, verify that the AI service is correctly configured as discussed in the previous training module. Identify the location of the zone configuration file as stated in the AI manifest file. Verify that the zone configuration file is accessible from the AI server and client. Verify that zone configuration is syntactically correct, that not collisions exist, and all stated resources are configured in the global zone.

Reviewing How Zones Work with IPS Commands, Package Publishers, and System Repository

- The use of IPS commands is basically the same for both the global zone and non-global zones.
- There is an important difference in the use of package publishers between the global zone and non-global zones.
 - Non-global zone: System repository provides access to the package repositories configured in the global zone
 - Global zone: Changes to the publisher configuration are seen immediately by all non-global zones via the system repository
- The zones proxy is a service that enables the package commands running inside a non-global zone to communicate with the system repository in the global zone.

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use most IPS commands in a non-global zone the same way you use them in the global zone. An important difference between the global zone and non-global zones is the use of package publishers. In a non-global zone, the system repository provides access to the package repositories configured in the global zone. Publisher configuration changes made to the global zone are seen immediately by all non-global zones via the system repository.

Note: The system repository will proxy http, https, and v4 file repositories and .p5p archive repositories.

The zones proxy is a service that enables `pkg` commands running inside a zone to communicate with the system repository, which is running in the global zone. The zones proxy has two parts. The `svc:/application/pkg/zones-proxyd:default` service runs in the global zone, and the `svc:/application/pkg/zones-proxy-client:default` service runs in the non-global zone.

Reviewing How Zones Work with IPS Software Package Installation and Updates

- Running install, update, or uninstall commands in the global zone:
 - With no arguments: Each non-global zone is updated
 - With package names specified: Only required changes are made to keep the non-global zone compatible with the global zone
- Changing facets and variants in the global zone can also affect non-global zones

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Installing, updating, and uninstalling packages in the global zone can affect the non-global zones. When you run the `pkg update` command with no arguments in the global zone, the global zone and each non-global zone is updated.

Note: Only a subset of the Oracle Solaris packages installed in the global zone are completely replicated when a non-global zone is installed. For example, many packages that contain the Oracle Solaris kernel are not needed in a non-global zone. All non-global zones implicitly share the same kernel from the global zone.

When you specify package names with the install, update, or uninstall commands in the global zone, IPS checks each non-global zone and makes changes only if required to keep the non-global zone compatible with the global zone.

Reviewing How Zones Work with IPS Software Package Installation and Updates

- Running package commands while logged in to a non-global zone only affects that non-global zone.
- In a non-global zone, you can:
 - Install different packages
 - Install different versions of the same package
 - Avoid different packages
 - Freeze packages at different versions
 - Select different default implementations
 - Set different facets
- In a non-global zone, the versions of installed packages can be restricted by the versions installed in the global zone.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you run package commands while logged into a non-global zone, only that non-global zone is affected. You can install different packages and install different versions of the same package if the result is compatible with the global zone. You can avoid different packages, freeze packages at different versions, set mediators to select different default implementations, and set different facets in the non-global zone image.

Versions of packages installed in a non-global zone can be restricted by the versions installed in the global zone. Some packages cannot be updated or downgraded in a non-global zone because those packages must be the same version in the non-global zone as they are in the global zone. For example, the package named entire must be the same in each non-global zone as in the global zone. The entire package constrains system package versions so that the resulting set of packages is a supportable image.

Reviewing How Zones Work with Physical Networks

- Zones communicate through IP network interfaces.
- The system administrator configures zone network interfaces during zone configuration.
- When a zone is booted, the network interfaces are set up and placed in the zone.
- Two IP types are available for non-global zones:
 - Shared-IP: A network interface is shared with the global zone
 - Exclusive-IP: A network interface is dedicated to the non-global zone

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

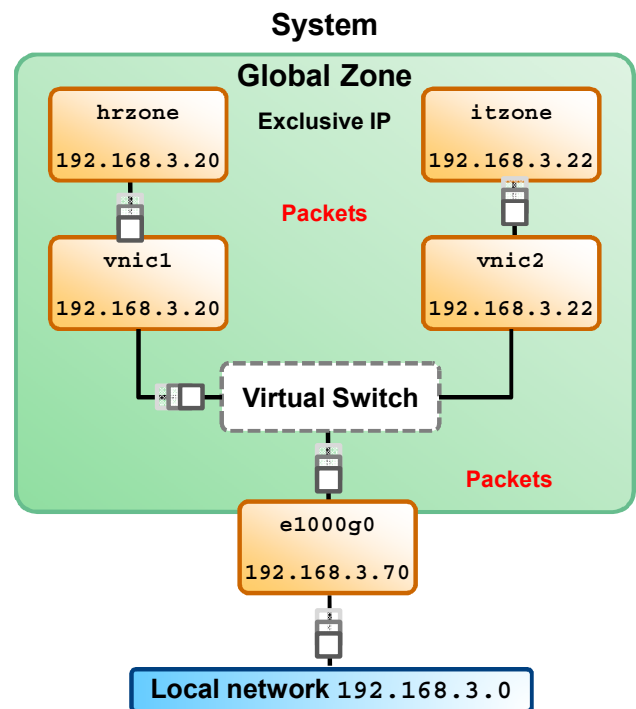
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Basic communication between zones is accomplished by giving each zone Internet Protocol (IP) network connectivity. An application running in one zone cannot observe the network traffic of another zone. This isolation is maintained even though the respective streams of packets travel through the same physical interface.

During zone configuration, the system administrator configures the zone network interfaces to provide network connectivity. These network interfaces are set up and placed in the zone when it is booted. Two IP types are available for non-global zones: shared-IP and exclusive-IP, which is the default. The shared-IP zones always share a network interface (or IP layer) with the global zone, and the exclusive-IP zones always have their own dedicated network interface (or instance of the IP layer). Both shared-IP zones and exclusive-IP zones can be used on the same machine.

Reviewing How Zones Work with Virtual Networks

- Use resource controls to:
 - Share bandwidth among VNICs
 - Customize link properties
 - Create flows



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Network virtualization is the process of consolidating hardware network resources and software network resources into a single virtual administrative unit. The end product of network virtualization is the *virtual network*. The goal of network virtualization is to provide systems and users with efficient, controlled, and secure sharing of the networking resources. A virtual network consists of one system using zones that are configured over at least one virtual network interface. The zones can communicate with each other as though on the same local network, providing a virtual network on a single host. The building blocks of the virtual network are virtual network interface cards or virtual NICs (VNICs) and virtual switches. Virtual networking provides a secure networking environment for your zone workloads. You can use virtual networking to consolidate network hardware and software. You can also use virtual networking to control bandwidth limits on selected zones.

When a virtual network is configured, a zone sends traffic to an external host in the same fashion as a system without a virtual network. Traffic flows from the zone, through the VNIC to the virtual switch, and then to the physical interface, which sends the data out onto the network.

To increase efficiency on your virtual network, you can implement controls to determine how resources are being used by the networking processes. The resource control features of Oracle Solaris enable bandwidth to be shared among the VNICs on a system's virtual network. Link properties that are specifically related to network resources, such as rings, CPUs, and so on, can be customized to process network packets. In addition, you can also create flows to manage network usage.

Quiz

Non-global zones installation errors during an AI install are sometimes the result of problems with the configuration information in the the global zone AI manifest.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Agenda

- Reviewing Zones Features and Functionality
- **Checking the Basics**
- Using Zones-related SMF Services
- Monitoring Non-Global Zones
- Fixing IPS Problems with Non-Global Zones
- Case Study 6

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Checking the Zones Configuration

- Displaying the current zone configuration on the system
- Determining the current zone configuration
- Displaying a zone configuration
- Displaying zone network information

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Displaying the Current Zones Configuration on the System

To display all running zones on the system, enter `zoneadm list`.

```
# zoneadm list
```

List options:

- c: Displays all configured zones
- i: Expands the display to all installed zones
- v: Displays verbose information, including zone name, ID, current state, root directory, brand type, IP-type, and options

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To display information about one or more zones on a system, you use the `zoneadm list` command.

Note: The `zoneadm` command is the primary tool used to administer non-global zones. Operations using the `zoneadm` command must be run from the global zone.

The `list` subcommand has several options. By itself, the `list` subcommand displays all running zones on the system.

To display all configured zones on the system, you can add the `-c` option. If you want to display all the installed zones on the system, you use the `-i` option. The `-v` option displays the zone name, ID, current state, root directory, brand type, IP-type, and options for the selected zone or zones.

You also have the option of combining options and even using all three options presented here simultaneously. For example, you can use `zoneadm list -civ` and `zoneadm list -cv` to show all the zones in any defined state: configured, incomplete, installed, running, shutting down, or down. The `zoneadm list -iv` command omits those zones that are only configured or incomplete. For more information about the `zoneadm` utility and its subcommands, see the `zoneadm(1M)` man page.

Determining the Current Zone Configuration

```
# zoneadm list -iv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
2	web	running	/zones/web	solaris	shared
4	storage	running	/zones/storage	solaris	shared
6	database	running	/zones/database	solaris	shared



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the output of `zoneadm list -iv`. On this system, there are four zones running. One is the global zone and three non-global zones: `web`, `storage`, and `database`. Each zone, including the global zone, is assigned a zone name. The global zone always has the name `global`. Each zone is also given a unique numeric identifier, which is assigned by the system when the zone is booted. The global zone is always mapped to ID 0.

Notice the paths for each zone. Each zone has a path to its root directory that is relative to the global zone's root directory. The global zone's path is root (`/`), and the paths for the non-global zones are `/zones/web`, `/zones/storage`, and `/zones/database` respectively. The brand for all the zones is `solaris`, which is the default zone brand in the Oracle Solaris 11.1 release. The global and non-global zones are set up as shared-IP.

Note: The `solaris` branded zone is on all supported sun4v and x86 architecture machines. The `solaris` branded zone uses the branded zones framework to run zones installed with the same software as is installed in the global zone. The system software must always be in sync with the global zone when using a `solaris` brand.

Displaying a Zone Configuration

To display a non-global zone configuration, use `zonecfg -z zonename info`.

```
# zonecfg -z web info
zonename: web
zonepath: /zones/web
brand: solaris
autoboot: true
bootargs:
file-mac-profile:
pool:
limitpriv:
scheduling-class:
ip-type: shared
hostid:
fs-allowed:
<output continued on next slide>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You must be the global administrator in the global zone, or a user with the correct rights profile, to display a zone's configuration. To display the configuration, you use the `zonecfg -z` command followed by the zone name and the `info` subcommand, as shown in this example.

The first part of the output displays the zone name (`web`), the zone path (`/zones/web`), the brand (`solaris`), and the setting of the `autoboot` option, which, when set to `true`, indicates that the zone should be booted automatically at system boot. Notice also the IP type setting. In this example, the IP type is `shared`, which, as you recall from the first topic, means that this non-global zone is sharing the IP layer with the global zone.

Displaying a Zone Configuration

```
<output continued from previous slide>
fs:
    dir: /opt/ora
    special: /opt/ora
    raw not specified
    type: lofs
    options: [ro]
net:
    address: 192.168.124.14/24
    allowed-address not specified
    configure-allowed-address: true
    physical: net0
    defrouter not specified
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The next section of the output (shown here) displays zone's file system information (`fs`), such as the directory location and the zone's network information, such as the IP address (192.168.124.14/24) and NIC (`net0`).

Note: If this zone were using a virtual network interface, after `physical` in the network section, you would see `vnic1` or `vnic2`.

Displaying Zone Network Information

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

```
# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
lo0/zoneadmd.v4	static	ok	127.0.0.1/8
lo0/zoneadmd.v4a	static	ok	127.0.0.1/8
lo0/zoneadmd.v4b	static	ok	127.0.0.1/8
net0/v4	static	ok	192.168.124.11/24
net0/zoneadmd.v4	static	ok	192.168.124.14/24
net0/zoneadmd.v4a	static	ok	192.168.124.15/24
net0/zoneadmd.v4b	static	ok	192.168.124.16/24
lo0/v6	static	ok	::1/128
lo0/zoneadmd.v6	static	ok	::1/128
lo0/zoneadmd.v6a	static	ok	::1/128
lo0/zoneadmd.v6b	static	ok	::1/128
net0/v6	addrconf	ok	fe80::214:4fff:fe02:5de6/10

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For network interface address information, you can use the `ipadm show-addr` command, as shown in this example. As you can see, it is more difficult to directly associate the zone with its IP address with this view. However, you can easily see the state of the interface.

Running DTrace in a Non-Global Zone

1. Use the `zonecfg limitpriv` property to add the `dtrace_proc` and `dtrace_user` privileges to the zone configuration, as follows:

```
global# zonecfg -z zonename
zonecfg:zonename> set \
limitpriv="default,dtrace_proc,dtrace_user"
zonecfg:my-zone> exit
```
2. Boot the zone by running `zoneadm -z zonename boot`.
3. Log in to the zone by running `zlogin zonename`.
4. Run the DTrace program by running `dtrace -l`.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

DTrace programs that require only the `dtrace_proc` and `dtrace_user` privileges can be run in a non-global zone. To add these privileges to the set of privileges available in the non-global zone, use the `zonecfg limitpriv` property, as shown in the step presented in the slide.

Note: The providers supported through `dtrace_proc` are `fasttrap` and `pid`. The providers supported through `dtrace_user` are `profile` and `syscall`. DTrace providers and actions are limited in scope to the zone.

Running DTrace in a Non-Global Zone: Example

```
global# zonecfg -z web
zonecfg:web> set limitpriv="default,dtrace_proc,dtrace_user"
zonecfg:web> exit
global# zoneadm -z web boot
global# zlogin web
global-web# dtrace -l
```

ID	PROVIDER	MODULE	FUNCTION NAME
1	dtrace		BEGIN
2	dtrace		END
3	dtrace		ERROR
1960	syscall		nosys entry
1961	syscall		nosys return
1962	syscall		rexit entry
1963	syscall		rexit return
1964	syscall		read entry
1965	syscall		read return
1966	syscall		write entry
1967	syscall		write return
1968	syscall		close entry
1969	syscall		close return
1970	syscall		linkat entry
1971	syscall		linkat return
1972	syscall		symlinkat entry
1973	syscall		symlinkat return

<output omitted>



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example in the slide, we are running DTrace in the `web` zone. In order to run DTrace in the zone, we add the `dtrace_proc` and `dtrace_user` privileges to the zone configuration by using the `zonecfg limitpriv` property. Next, we boot the zone. We then log in to the zone and run the DTrace program.

Shutting Down and Starting Up a Non-Global Zone

To shut down a zone, use `zoneadm -z zonename shutdown -i 0`.

```
global# zoneadm -z web shutdown -i 0
```

To start a zone, use `zoneadm -z zonename boot`.

```
global# zoneadm -z web boot
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are various reasons why you might need to shut down a zone. For example, another zone needs to complete a database update before the zone you have been told to shut down can be brought back up.

To shut down a non-global zone you must be the global administrator or a user with appropriate authorizations in the global zone. This procedure is used to cleanly shut down a zone (as opposed to halting a zone).

To shut down a zone, from the global zone you use the `zoneadm -z` command followed by the zone name and specify `shutdown` as the command to run and `init 0` as the state, as shown in the example.

Note: At this time, you cannot use the `shutdown` command to place the zone in single-user state.

After a zone has been shut down, you can start it again by using the `zoneadm -z` command followed by the zone name and `boot`, as shown in this example. When a zone is booted, the required services and the facilities for that zone are brought online. These services help the zone to be functional and ready to use.

Halting a Non-Global Zone

To halt a zone, run `zoneadm -z zonename halt`.

```
global# zoneadm -z web halt
```

To verify that the zone has been halted, run `zoneadm list -iv`.

```
global# zoneadm list -iv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	web	installed	/zones/web	solaris	shared
4	storage	running	/zones/storage	solaris	shared
6	database	running	/zones/database	solaris	shared

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When you need to remove a zone, you use the `zoneadm -z halt` command with the zone name, as shown in the example. When halted, the zone is brought back to the installed state.

To verify that the zone has been halted and is no longer running, you can run the `zoneadm list -iv` command, as shown in the second example. As you can see in the output, only the global zone and the other two non-global zones are running.

Note: Although you can halt a zone, the recommended way to bring a zone down is by using the `zoneadm shutdown` command. This approach brings the zone down more gently.

Troubleshooting Zones That Do Not Halt

- Problem: The zone in an intermediate state, somewhere between running and installed
- Most common cause of failure: Inability of the system to unmount all file systems
- Corrective action: Use the tools described in the `proc(1)` (see `pfiles`) and `fuser(1M)` man pages to:
 - Ensure processes are stopped
 - Ensure files are unmounted

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the event that the system state associated with the zone cannot be destroyed, the halt operation will fail halfway. This leaves the zone in an intermediate state, somewhere between running and installed. In this state there are no active user processes or kernel threads, and none can be created. When the halt operation fails, you must manually intervene to complete the process.

The most common cause of a failure is the inability of the system to unmount all file systems. Unlike a traditional Oracle Solaris system shutdown, which destroys the system state, zones must ensure that no mounts performed while booting the zone or during zone operation remain once the zone has been halted. Even though `zoneadm` makes sure that there are no processes executing in the zone, the unmount operation can fail if processes in the global zone have open files in the zone. Use the tools described in the `proc(1)` (see `pfiles`) and `fuser(1M)` man pages to find these processes and take appropriate action. After these processes have been dealt with, reinvoking `zoneadm halt` should completely halt the zone.

Uninstalling a Non-Global Zone

1. List the zones on the system by using the `zonedadm list -iv` command.
2. If the zone is in the running state, shut it down by using the `zoneadm -z zonename shutdown -i 0` command.
3. Run the `zonedadm list -iv` command again to verify that the zone is in the installed state.
4. Run the `zoneadm -z zonename uninstall` command to remove the zone.
5. Verify that the uninstalled zone is no longer listed by using the `zonedadm list -iv` command.

Caution: The action of removing all of the files in the zone's root file system is irreversible.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There might be reasons why you need to uninstall a zone. Whatever the reason, you should always use caution when uninstalling a zone because the effects of these actions are irreversible.

The steps for uninstalling a zone are presented in the slide.

Note: You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

Note for step 2: The uninstall operation is invalid for running zones.

Note for step 4: You can also use the `-F` option to force the action. If you do not specify this option, the system will prompt you for confirmation.

Troubleshooting Zones That Do Not Uninstall

- Problem #1: The zone is left in the incomplete state.

```
# zoneadm list -cv
ID NAME      STATUS      PATH                                BRAND  IP
0  global    running     /                                  solaris shared
2  storage   running     /zones/storage                     solaris shared
3  database  running     /zones/database                     solaris shared
-  web       incomplete  /zones/web                          solaris shared
```

- Corrective action: Run the `zoneadm uninstall -F` command.

```
# zoneadm -z web uninstall -F
Progress being logged to
/var/log/zones/zoneadm.20000207T165138Z.web.uninstall
promoting 'rpool/zones/web/rpool/ROOT/solaris-1/var'.
promoting 'rpool/zones/web/rpool/ROOT/solaris-1'.
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If a zone uninstall is interrupted, the zone is left in the incomplete state. Use the `zoneadm uninstall -F` command to reset the zone to the configured state.

Troubleshooting Zones That Do Not Uninstall

- Problem #2: The `zonepath` is not removed.

```
# zonecfg -z web delete
Are you sure you want to delete zone web (y/[n])? y
# ls /zones
database  storage  web
```

- Restoring a backup of the zone to the original path fails.

```
# zonecfg -z web -f /var/web.cfg
# zoneadm -z web attach -a /var/web
...
Progress being logged to
  /var/log/zones/zoneadm.20000207T191107Z.web.attach
Installing: This may take several minutes...
Received dataset ROOT/solaris-1 collides with existing dataset
  rpool/zones/web/rpool/ROOT/solaris-1.
...
Result: Attach Failed.
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If the `zonepath` is not removed, this could be an indication that this zone is installed in another boot environment (BE). The `zonepath` and various datasets that exist within the `zonepath` dataset are not removed while a boot environment exists that has an installed zone with a given `zone path`. This presents a problem if the administrator created a backup of the zone and wishes to restore the zone to its original `zone path`. The zone attach operation will fail due to dataset collisions.

Troubleshooting Zones That Do Not Uninstall

- Corrective actions:
 - If possible, remove old BEs before uninstalling the zone.

```
# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      NR      /           10.12G static 2000-01-02 21:45
solaris.orig -      -           1.19M static 2000-01-02 23:52
# beadm destroy -F solaris.orig
# zoneadm -z web uninstall -F
# zoneadm -z web attach -a /var/web
```

- If the path name and datasets are present after uninstalling the zone, manually remove the datasets.

```
# zfs destroy -r rpool/zones/web
# zoneadm -z web attach -a /var/web
...
Updating non-global zone: Zone updated.
Result: Attach Succeeded.
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The corrective action is to remove the old BEs before you attempt to remove the zone. If this is not possible or you forget to remove the old BEs and you uninstall the zone, the `zone` path and datasets will not be removed. In this case, you can manually remove the zone's datasets (indicated in the zone attach log as collisions) and then proceed with the zone restore operation.

Cloning a Non-Global Zone

1. Halt the source zone to be cloned by using the `zoneadm -z zonename halt` command.
2. Start configuring the new zone by exporting the configuration of the source zone to a file by using the `zonecfg -z zonename export -f /zones/filename` command.
3. Edit the file to set different properties and resources.
4. Create the new zone by using the commands in the edited file: `zonecfg -z newzone -f /zones/filename`
5. Install the new zone by cloning the source zone: `zoneadm -z newzone clone zonename`
6. Verify that the cloned zone has been installed by using the `zonedadm list -iv` command.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

[SMEs: What are some reasons for needing to clone a zone as part of troubleshooting?]

The steps for cloning a zone are presented in the slide.

Note: You must be the global administrator or a user with appropriate authorizations in the global zone to perform this procedure.

Note for step 3: Set different properties and resources for the components that cannot be identical for different zones. For example, you must set a new `zonepath`. For a shared-IP zone, the IP addresses in any net resources must be changed. For an exclusive-IP zone, the physical property of any net resource must be changed.

Note for step 6: If a storage object contains any preexisting partitions, zpools, or UFS file systems, the clone fails and an error message is displayed. To continue the operation and overwrite any preexisting data, use the appropriate `-x` option to `zoneadm clone`. The source zone must be uninstalled before the `force` subcommand can be used.

```
-x force-zpool-import
-x force-zpool-create=zpoolname
-x force-zpool-create=zpoolname1, zpoolname2, zpoolname3
-x force-zpool-create-all
```

Cloning a Non-Global Zone Error

```
# zoneadm list -cv
ID NAME          STATUS  PATH                                BRAND
IP
0 global         running /                                solaris shared
2 storage        running /zones/storage                  solaris shared
3 database        running /zones/database                 solaris shared
- web            installed /zones/web                     solaris shared
- web1           configured /zones/web                     solaris shared

# zoneadm -z web1 clone web
WARNING:network address '192.168.124.24' is configured in both zones.
web1 zonepath(/zones/web) and web zonepath (/zones/web overlap.
zoneadm: zone 'web1': could not verify zonepath /zones/web because of
the above errors.
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This example shows common mistake made by administrators when cloning a zone. In this case, the administrator created the clone target configuration from a source zone named `web` (`zonecfg -z web export -f /var/tmp/web.cfg`) and then used that configuration file to place the target zone named `web1` in the configured state (`zonecfg -z web1 -f /var/tmp/web.cfg`). But when attempting to clone the `web` zone to the `web1` zone, the cloning operation failed. The failure is caused by a duplicate (overlapping) `zonepath` property in both the source and target zone configurations. The administrator should have updated the exported `web` configuration file with a unique `zonepath` property before attempting to clone the `web1` zone.

Quiz

The `zonecfg limitpriv` property is used to add the `dtrace_proc` and `dtrace_user` privileges to the zone configuration.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Agenda

- Reviewing Zones Features and Functionality
- Checking the Basics
- **Using Zones-Related SMF Services**
- Monitoring Non-Global Zones
- Fixing IPS Problems with Non-Global Zones
- Case Study 6

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Identifying the Zones-Related SMF Services and Their Roles

- Global zone-related SMF services:
 - Starts each zone that has `autoboot=true`:
`svc:/system/zones:default`
 - Performs zone installation on first boot, if needed:
`svc:/system/zones-install:default`
 - Used by the packaging system to provide zones access to the system repository: `svc:/application/pkg/zones-proxyd:default`
 - Caching proxy server that caches `pkg` data and metadata used during zone installation and other `pkg` operations. See the `pkg(1)` and `pkg(5)` man pages:
`svc:/application/pkg/system-repository:default`
 - Controls `zonestatd`: `svc:/system/zones-monitoring:default`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide lists the zones-related SMF services in the global zone.

Identifying the Zones-Related SMF Services and Their Roles

- Non-global zones-related SMF service:
 - Used by the packaging system to provide zones access to the system repository: `svc:/application/pkg/zones-proxy-client:default`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide displays the non-global zones-related SMF service.

Using SMF Troubleshooting Techniques Within a Non-Global Zone

To check the status of SMF services in a non-global zone from the global zone, run `zlogin zonename svcs -a`.

```
global# zlogin web svcs -a
STATE          STIME          FMRI
legacy_run     12:13:17      lrc:/etc/rc2_d/S89PRESERVE
disabled       12:12:49      svc:/network/ipsec/ike:default
<output omitted>
```

To check the status of SMF services from within a non-global zone, log in to the zone and then run `svcs -a`.

```
global# zlogin web
[Connected to zone 'web' pts/2]
Oracle Corporation SunOS 5.11 11.1   September 2012
global-web:~# svcs -a
STATE          STIME          FMRI
legacy_run     12:13:17      lrc:/etc/rc2_d/S89PRESERVE
disabled       12:12:49      svc:/network/ipsec/ike:default
<output omitted>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can check the status of the SMF services running in a non-global zone from either the global zone, as shown in the first example, or from within a non-global zone, as shown in the second example.

[SMEs: What else do we want to say about this “technique”.]

Quiz

Which SMF service is used by the packaging system to provide non-global zones access to the system repository?

- a. `svc:/system/zones-monitoring:default`
- b. `svc:/application/pkg/zones-proxy-client:default`
- c. `svc:/application/pkg/system-repository:default`
- d. `svc:/application/pkg/zones-proxyd:default`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Agenda

- Reviewing Zones Features and Functionality
- Checking the Basics
- Using Zones-related SMF Services
- **Monitoring Non-Global Zones**
- Fixing IPS Problems with Non-Global Zones
- Case Study 6

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Using zonestat

To determine a zone's resource utilization, use the `zonestat` utility.

```
# zonestat -r summary 5
root@host01:~# zonestat -r summary 5
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
SUMMARY          Cpus/Online: 24/24   PhysMem: 15.8G   VirtMem: 17.8G
---CPU---  --PhysMem--  --VirtMem--  --PhysNet--
      ZONE  USED  %PART  USED  %USED  USED  %USED  PBYTE  %PUSE
[total]  0.11  0.46%  2234M  13.7%  4358M  23.8%    0  0.00%
[system]  0.02  0.12%  1290M  7.93%  3426M  18.7%    -    -
  global  0.07  0.33%   672M  4.13%   589M  3.21%    0  0.00%
 database 0.00  0.00%   96.8M  0.59%   123M  0.67%    0  0.00%
  storage 0.00  0.00%   96.2M  0.59%   121M  0.66%    0  0.00%
    web   0.00  0.00%   79.2M  0.48%   97.1M  0.53%    0  0.00%
...
...
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you are asked to monitor a zone's resource utilization, you can do so with the `zonestat` utility. The `zonestat` utility provides highly accurate reports on the CPU, memory, and resource control utilization of running zones. Each zone's utilization is reported as a percentage of both system resources and the zone's configured limits. You can specify the interval at which you want the utility to print a report. In the example shown here, you have selected one interval with a duration of 5 seconds.

The `zonestat` utility can be used from within a zone, for a localized view of zone system resources, or from the global zone for a system-wide view of zone system resources. With this utility, it is very easy to identify resource bottlenecks or misbehaving applications. The system administrator is able to take prompt remedial action, accurately addressing the identified problem. Capacity planning is also greatly simplified. In addition, `zonestat` also helps the administrator quickly understand how zones and resource management have been configured. This can be particularly useful when taking over administration duties on an unfamiliar system.

One of the uses of the `zonestat` utility is to monitor periodic usage of the system's physical memory, virtual memory, and CPU resources. In the example shown here, we are monitoring the resource utilization of all the zones from the global zone by using the `zonestat` command followed by `-r summary 5`. The output shows how many CPUs are available on the system and how many are being used. If a CPU has been partitioned (that is, if multiple processor sets have been created on the system by dividing up the CPU), usage of the partition is displayed.

To the right of the CPU usage, the physical and virtual memory usage by the system and by each zone are displayed. Physical memory refers to RAM on the machine, and virtual memory refers to RAM plus system swap space. The memory usage shows how much memory is being used and the percentage of memory being used. To the right of the virtual memory is the physical network utilization for each zone. You can use this information to determine the load on an NIC.

Note: Oracle Solaris enables you to regulate memory consumption in nonglobal zones by using a resource capping daemon. For more information on resource capping, see "Physical Memory Control Using the Resource Capping Daemon" in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

For more information about the `zonestat` utility, see the `zonestat(1)` man page.

Using fsstat

- -z option: Specifies the zone or zones to report
- -Z option: Reports per-zone activity for each zone
- -A flag: Reports aggregate activity for the specified fstypes across all examples

```
# fsstat -A -z web -z storage tmpfs zfs
new  name    name    attr  attr  lookup rddir  read read  write write
file remov  chng    get   set   ops   ops   ops bytes ops bytes
192K 159K 32.2K 1.46M 1.45K 425K 344 5.20M 5.29G 2.67M 4.17G tmpfs
2.49K 2.37K 93 19.6K 26 3.92K 0 66.4K 74.3M 42.3K 54.8M tmpfs:web
44.2K 12.9K 11.8K 5.12M 86.8K 19.9M 189K 5.73M 10.9G 439K 7.54G zfs
66 53 21 163K 95 506K 3.89K 173K 223M 3.61K 40.1M zfs:web
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Previous to the Oracle Solaris 11.1 release, a zone administrator could not monitor activity in a particular zone, nor could a non-global zone administrator see the file system activity in his own zone. Now the `fsstat` utility has been enhanced with a per-fstype `kstat` for each zone, and the global zone has a `kstat` reporting its exclusive activity.

The `-z` option allows you to specify the zone or zones to report, and the `-Z` option reports per-zone activity for each zone. Also, the new `-A` flag reports aggregate activity for the specified file system types, which is the default behavior if neither `-z` nor `-Z` are specified. If you use these in conjunction with each other, the aggregate is additionally reported for the specified `fstypes` across all zones.

In the example presented in the slide, the stats for zones `web` and `storage` are displayed as well as the system-wide aggregate for file system types `tmpfs` and `zfs`.

Note: The `-z`, `-Z`, and `-A` options have no effect if not used to monitor file system types.

For more information about the enhancements made to this utility, see the `fsstat` man page.

Quiz

You specifically want to monitor the file system type `zfs` activity in `zone1`. Which command would you use to do this?

- a. `fsstat -A -Z tmpfs zfs`
- b. `fsstat -A -Z zone1 -z zone2 tmpfs zfs`
- c. `fsstat -z global -F`
- d. `fsstat -z zone1 zfs`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Agenda

- Reviewing Zones Features and Functionality
- Checking the Basics
- Using Zones-related SMF Services
- Monitoring Non-Global Zones
- **Fixing IPS Problems with Non-Global Zones**
- Case Study 6

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Analyzing Package Version Problems Between a Non-Global Zone and the Global Zone

Linked Images Behavior

- Most `pkg` operations cause non-global zones to be synced.
- Installing a package in the global zone does not install it in all the non-global zones.
- Uninstalling a package in the global zone does not uninstall it in all non-global zones.
- Updating a package in the global zone does the minimal update in the non-global zones needed to keep them in sync.
 - Exception: Using `pkg update` with no arguments updates each non-global zone as far forward as possible.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When analyzing package version problems between a non-global zone and the global zone, you need to keep the following information in mind. Oracle Solaris 11.1 has introduced a new IPS feature called Linked Images. This feature behaves in the following way. Most `pkg` operations cause the non-global zones to be synced. However, installing a package in the global zone does not install it in all the non-global zones. The same applies to uninstalling a package in the global zone. Updating a package in the global zone does a minimal update to the non-global zone to keep it in sync, with one exception: The `pkg update` command with no arguments will update each non-global zone as far forward as it can.

Analyzing Package Version Problems Between a Non-Global Zone and the Global Zone

Linked Images Known Issues

- Non-global zones are updated serially, which means the update operation can be slow.
- Using the `pkg` command with an alternate root (the `-R` option) on systems with zones does not really work unless the publishers in the alternate root image exactly match the publishers in the booted image.
- Using `-g` with the `install` and `update` commands does not work if non-global zones exist.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are some known issues with the Linked Images feature. The first being that, by default, non-global zones are updated serially, which means the update operation can be slow, especially if there are a lot of zones to update.

Note: Zones can be configured to be updated in parallel instead of serially. The parallel update provides a significant improvement in the time required to update all the zones on a system. To update multiple non-global zones concurrently, use the `-C` option or set the `PKG_CONCURRENCY` environment variable in the global zone. The `-C n` option and the `PKG_CONCURRENCY=n` environment variable specify to update at most *n* images in parallel for *n* greater than or equal to 1. The default value of *n* is 1. If *n* is 0 or a negative number, all non-global zones are updated in parallel with the global zone. The `PKG_CONCURRENCY` environment variable is ignored if the `-C` option is specified. The `-C` option and the `PKG_CONCURRENCY` environment variable can be used with `pkg install`, `pkg uninstall`, `pkg change-variant`, and `pkg change-facet` as well as with `pkg update`.

Also, if you have non-global zones, using the `pkg update` command with the `-R` option doesn't work unless the publishers in the alternate root image match the ones in the booted image. Finally, using the `-g` option with the `install` and `update` commands to specify a temporary source of packages does not work if non-global zones exist.

Verifying the Zones Services Are Online

- Make sure the three SMF services are online:
 - `svc:/application/pkg/system-repository:default`
 - `svc:/application/pkg/zones-proxyd:default`
 - `svc:/application/pkg/zones-proxy-client:default`
- Make sure there is communication between the non-global zones, the global zone, and the publisher.
- Check that the zone is pointing to the expected publishers.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When troubleshooting the package problems with zones, there are several things you will want to check. First, remember to check that the three SMF services for the proxies are online by using the techniques we discussed in the topic on Using Zones-Related SMF Services. You'll recall from our earlier discussion that the first two services presented in the slide run in the global zone. The third service runs in the non-global zones. The system repository uses these services to share publisher configuration information to all the non-global zones on the system as well as to the publishers configured in the global zone.

Notes:

For more information about the system repository and zones proxy services, see the `pkg.sysrepo(1M)` man page.

Next, make sure there is communication between the non-global zones, the global zone, and the publisher by using the `pkg publisher` command and verifying that the publisher is online.

You cannot reconfigure the system repository from within a non-global zone. For example, you cannot change the origins or properties of publishers or the publisher search order of publishers whose location is `<system-repository>`. If you cannot reach a publisher, you can set a proxy in the global zone by setting the `http_proxy` environment variable or by specifying the `--proxy` option to the `pkg set-publisher` command.

You also want to make sure that the zone is pointing to the expected publishers.

Verifying Communication Between the Zones and the Publisher: Example

```
global:~# pkg publisher
PUBLISHER      TYPE      STATUS P LOCATION
solaris         origin    online F http://pkg.oracle.com/solaris/release/
solaris         origin    online F file:///net/192.168.124.1/seif/os/s11_u1_repo/
ouif            origin    online F file:///net/192.168.124.1/seif/Solaris/s11_ouif_repo/
```

```
global:~# zlogin web
[Connected to zone 'web' pts/2]
Oracle Corporation  SunOS 5.11  11.1 September 2012
global-web:~# pkg publisher
PUBLISHER      TYPE      STATUS P LOCATION
solaris (syspub)  origin    online T <system-repository>
solaris (syspub)  origin    online F <system-repository>
ouif (syspub)    origin    online F <system-repository>
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the first example in the slide, we are displaying the publishers for the global zone. We want to verify that the publishers are online and communicating with the global zone, which they are.

Note: The P column specifies whether the location is proxied. Values in this column are true (T) or false (F). Use either of the following commands to get more information: `pkg publisher -F tsv` or `pkg publisher solaris`.

The second example shows how these same publishers appear when you are logged into a non-global zone. Again, check that the publisher is online and also verify that the zone is pointing to the expected publisher.

Choosing Different Origins That Are Configured in the Global Zone

- Try connecting through the system repository in the global zone.
- Choose various origins that are configured in the global zone.
 - HTTP origins:
`http_proxy=http://localhost:1008 curl`
`http://ipkg.us.oracle.com/solaris11/dev/publisher/0`
 - File origins:
`curl http://localhost:1008/nss/<hash>/publisher/0`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Another probable cause of package problems in a zone is the origin configuration. To test this possibility, try connecting through the system repository in the global zone and through various origins that have been configured in the global zone.

Let's take a moment to review the different origin configurations you might encounter. Access to repositories configured in the global zone is provided to non-global zones using the `system-repository` service. Any updates to proxies for origins in the global zone are automatically made to the `system-repository` configuration. Using this method, no modifications are required to the `system-repository` SMF service.

It is also possible to configure the proxies used by the `system-repository` SMF service, overriding any proxies configured on publishers in the global zone. The `system-repository` proxies can be set by using the `config/http_proxy` or `config/https_proxy` SMF properties.

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

You can configure the proxy directly in the global zone, and any updates to proxies for origins in the global zone are automatically made to the `system-repository` configuration. The `system-repository` service does not require modification. No port specification is required unless the proxy accepts connections on a port other than 80. If zones are on the system, the `system-repository` service is restarted, and the proxy is used to provide access to *pkg-server*.

Checking the Zone State

Zone State	Effect on Package Operations
Configured	Package tools can be run. No software has been installed yet.
Incomplete	If <code>zoneadm</code> is operating on the zone, package tools should not be used. If no <code>zoneadm</code> process is operating on the zone, package operations are safe to run, but no software in this zone will be changed and any software in the zone will not affect dependency resolution.
Unavailable	The software image within the zone is not accessible. The software image will not be changed, nor will it affect dependency resolution.
Installed	Package tools can be run. Note that immediately after <code>zoneadm -z zonename install</code> has completed, the zone is also moved to the installed state.
Ready	Package tools can be run.
Running	Package tools can be run.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Another item worth checking is the zone state. The table presented in the slide describes how a zone state affects package operations.

Note for the Unavailable Zone State: A non-global zone transitions to the unavailable state when the storage for the zone is not accessible or when the image of the zone is out of sync with the global zone's image. This state transition occurs to prevent a problem that is affecting a non-global zone from blocking package operations in the global zone. When a zone's storage is temporarily unavailable and package operations that change the version of installed software occur, it is likely that after fixing the storage problem, the zone might need to be attached by using one of the `solaris` brand's `attach` options that allow for updates. For example, `zoneadm -z zonename attach -u` might be required to synchronize versions of critical software between the global zone and a non-global zone that is in the unavailable state.

Agenda

- Reviewing Zones Features and Functionality
- Checking the Basics
- Using Zones-related SMF Services
- Monitoring Non-Global Zones
- Fixing IPS Problems with Non-Global Zones
- **Case Study 6**

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 6: Zone Fails to Attach!

- Clarify problem statement.
 - The `web` zone in system `host2` fails to restore.
- Using “assumptive” method, prioritize the most probable causes of failure.
 - Corrupted or incomplete zone backup
 - Collision with existing datasets in zone path
- Starting with the most probable cause of failure, gather pertinent information.
 - Replicate the error.
 - Peruse zone attach log.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 6: Troubleshooting Steps

The follows steps are used to identify the cause of the problem:

1. Check zone backup files.
2. Verify that the zone restore operation fails.
3. Search the search attach log for error messages (if available).
4. Determine if alternate BE exists.
5. Check zone path for existing datasets
6. Repair the fault.
7. Verify the solution.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the high-level steps involves in this case study.

Case Study 6: Walkthrough

```
##### Check the web zone backup files. #####
# cat /var/backups/zone_configs/web.cfg
create -b
set brand=solaris
set zonepath=/zones/web
set autoboot=true
set ip-type=shared
add fs
set dir=/opt/ora
set special=/opt/ora
set type=lofs
add options ro
end
add net
set address=192.168.124.24/24
set configure-allowed-address=true
set physical=net0
End
# file /var/backups/zones/web
web: ZFS snapshot stream
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide begin the Case Study 4 walkthrough. Note that line beginning with "#####" are comments.

Case Study 6: Walkthrough

```
##### Configure the web zone. #####
# zonecfg -z web -f /var/backups/zone_configs/web.cfg
# zonecfg -z web info
zonename: web
zonepath: /zones/web
brand: solaris
autoboot: true
bootargs:
file-mac-profile:
pool:
limitpriv:
scheduling-class:
ip-type: shared
hostid:
fs-allowed:
fs:
dir: /opt/ora
special: /opt/ora
raw not specified
type: lofs
options: [ro]
net:
address: 192.168.124.24/24
allowed-address not specified
configure-allowed-address: true
physical: net0
defrouter not specified
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right of the "E".

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 6: Walkthrough

```
##### Attach the web zone backup snapshot. #####
# zoneadm -z web attach -a /var/backups/zones/web
Warning: The -a and -d options to the attach subcommand may be removed in a
future release of Solaris. Use of the install subcommand is recommended.
Progress being logged to /var/log/zones/zoneadm.20130328T232100Z.web.attach
Installing: This may take several minutes...
Received dataset ROOT/solaris-1 collides with existing dataset
    rpool/zones/web/rpool/ROOT/solaris-1.
Received dataset ROOT/solaris-1/var collides with existing dataset
    rpool/zones/web/rpool/ROOT/solaris-1/var.
Received dataset VARSHARE collides with existing dataset rpool/zones/web/rpool/VARSHARE.
Received dataset export collides with existing dataset rpool/zones/web/rpool/export.
Received dataset export/home collides with existing dataset
    rpool/zones/web/rpool/export/home.
Received dataset export/home/oracle collides with existing dataset
    rpool/zones/web/rpool/export/home/oracle.
ERROR: Error: Command <zfs destroy rpool/zones/web/installtmp> exited with status 1
ERROR: Error: cannot destroy dataset rpool/zones/web/installtmp
ERROR: Invalid data received
Result: Attach Failed.
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right of the "E".

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 6: Walkthrough

```
##### Look for errors in the zone attach log. #####
# more /var/log/zones/zoneadm.20130328T232100Z.web.attach
[Thursday, March 28, 2013 05:21:00 PM MDT] ==== Starting: /usr/lib/brand/solaris
/attach web /zones/web -a /var/backups/zones/web ====
...
[Thursday, March 28, 2013 05:22:31 PM MDT] Mounting rpool/zones/web/rpool/ROOT/s
olaris-1 at /tmp/tmp.rRa4Pu with ZFS temporary mount
[Thursday, March 28, 2013 05:22:32 PM MDT] Received dataset ROOT/solaris-1 colli
des with existing dataset rpool/zones/web/rpool/ROOT/solaris-1.
[Thursday, March 28, 2013 05:22:32 PM MDT] Mounting rpool/zones/web/rpool/ROOT/s
olaris-1/var at /tmp/tmp.wbaqRu with ZFS temporary mount
[Thursday, March 28, 2013 05:22:33 PM MDT] Received dataset ROOT/solaris-1/var c
ollides with existing dataset rpool/zones/web/rpool/ROOT/solaris-1/var.
[Thursday, March 28, 2013 05:22:33 PM MDT] Mounting rpool/zones/web/rpool/VARSHA
RE at /tmp/tmp.FyaWSu with ZFS temporary mount
[Thursday, March 28, 2013 05:22:35 PM MDT] Received dataset VARSHARE collides wi
th existing dataset rpool/zones/web/rpool/VARSHARE.
[Thursday, March 28, 2013 05:22:35 PM MDT] Mounting rpool/zones/web/rpool/export
at /tmp/tmp.KVaiUu with ZFS temporary mount
[Thursday, March 28, 2013 05:22:36 PM MDT] Received dataset export collides with
existing dataset rpool/zones/web/rpool/export.
[Thursday, March 28, 2013 05:22:36 PM MDT] Mounting rpool/zones/web/rpool/export
/home at /tmp/tmp.ZcaGVu with ZFS temporary mount
[Thursday, March 28, 2013 05:22:37 PM MDT] Received dataset export/home collides
with existing dataset rpool/zones/web/rpool/export/home.
[Thursday, March 28, 2013 05:22:37 PM MDT] Mounting rpool/zones/web/rpool/export
/home/oracle at /tmp/tmp.2va4Wu with ZFS temporary mount
[Thursday, March 28, 2013 05:22:38 PM MDT] Received dataset export/home/oracle c
ollides with existing dataset rpool/zones/web/rpool/export/home/oracle.
cannot destroy 'rpool/zones/web/installtmp': filesystem has children.
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 6: Walkthrough

```
use '-r' to destroy the following datasets:  
rpool/zones/web/installtmp/ds@snap  
rpool/zones/web/installtmp/ds/rpool@snap  
rpool/zones/web/installtmp/ds/rpool/export@snap  
rpool/zones/web/installtmp/ds/rpool/export/home@snap  
rpool/zones/web/installtmp/ds/rpool/export/home/oracle@snap  
rpool/zones/web/installtmp/ds/rpool/export/home/oracle  
rpool/zones/web/installtmp/ds/rpool/export/home  
rpool/zones/web/installtmp/ds/rpool/export  
rpool/zones/web/installtmp/ds/rpool/ROOT@snap  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris-1@snap  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris-1/var@snap  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris-1/var  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris-1  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris@2013-03-20-03:28:31  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris@snap  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris/var@2013-03-20-03:28:31  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris/var@snap  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris/var  
rpool/zones/web/installtmp/ds/rpool/ROOT/solaris  
rpool/zones/web/installtmp/ds/rpool/ROOT  
rpool/zones/web/installtmp/ds/rpool/VARSHARE@snap  
rpool/zones/web/installtmp/ds/rpool/VARSHARE  
rpool/zones/web/installtmp/ds/rpool  
rpool/zones/web/installtmp/ds  
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 6: Walkthrough

```
##### Determine if alternate BE (s) exist. #####
# beadm list
BE           Active Mountpoint Space   Policy Created
--           -
solaris      NR           /           11.18G static 2013-03-19 19:07
solaris.orig -           -           96.75M static 2013-03-19 21:28
##### Determine if datasets exist in the web zone zonepath. #####
root@host02:~# zfs list |grep web
rpool/zones/web                               479M 42.7G 35K /zones/web
rpool/zones/web/rpool                         479M 42.7G 31K /rpool
rpool/zones/web/rpool/ROOT                    478M 42.7G 31K legacy
rpool/zones/web/rpool/ROOT/solaris-1          478M 42.7G 453M /
rpool/zones/web/rpool/ROOT/solaris-1/var      24.9M 42.7G 24.9M /var
rpool/zones/web/rpool/VARSHARE                 58K 42.7G 39K /var/share
rpool/zones/web/rpool/export                  138K 42.7G 32K /export
rpool/zones/web/rpool/export/home             85.5K 42.7G 32K /export/home
rpool/zones/web/rpool/export/home/oracle      33.5K 42.7G 33.5K

##### Choose a solution: delete the alternate BE or manually delete the datasets. #####
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 6: Walkthrough

```
##### In this example, we destroy the datasets. #####
# zfs destroy -r rpool/zones/web
# zfs list |grep web
##### Attach the web zone. #####
# zoneadm -z web attach -a /var/backups/zones/web
The following ZFS file system(s) have been created:
    rpool/zones/web
Warning: The -a and -d options to the attach subcommand may be removed in a
future release of Solaris. Use of the install subcommand is recommended.
Progress being logged to /var/log/zones/zoneadm.20130328T235102Z.web.attach
    Installing: This may take several minutes...
        Zone BE root dataset: rpool/zones/web/rpool/ROOT/solaris-0
        Cache: Using /var/pkg/publisher.
    Updating non-global zone: Linking to image /.
Processing linked: 1/1 done
    Updating non-global zone: Auditing packages.
No updates necessary for this image.

    Updating non-global zone: Zone updated.
        Result: Attach Succeeded.
Log saved in non-global zone as
    /zones/web/root/var/log/zones/zoneadm.20130328T235102Z.web.attach
# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	engineering	running	/zones/engineering	solaris	excl
3	storage	running	/zones/storage	solaris	shared
4	database	running	/zones/database	solaris	shared
-	web	installed	/zones/web	solaris	shared



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Additional Resources

- OTN Oracle Solaris 11.1 Virtualization Technology:
<http://www.oracle.com/technetwork/server-storage/solaris11/technologies/virtualization-306056.html>
- OTN Oracle Solaris 11.1 Networking Virtualization Technology Page (white papers, blogs, etc.):
<http://www.oracle.com/technetwork/server-storage/solaris11/technologies/networkvirtualization-312278.html>
- Documentation: Oracle Solaris 11.1 Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management:
http://docs.oracle.com/cd/E26502_01/html/E29024/index.html

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 7: Overview

- 7-1: Troubleshooting a zone startup failure
- 7-2: Troubleshooting a zone permissions issue
- 7-3: Troubleshooting a zone restore failure

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

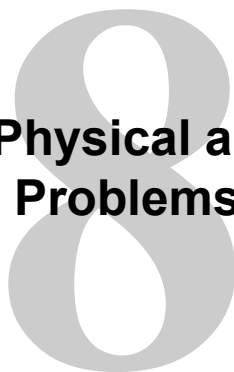
In this lesson, you should have learned how to:

- Recall the zones features and functionality
- Verify the zone configuration
- Shut down a non-global zone
- Halt a non-global zone
- Uninstall a non-global zone
- Clone a non-global zone
- Troubleshoot a zone that does not halt
- Identify zones-related SMF services
- Use SMF troubleshooting techniques within a zone
- Monitor zones
- Fix IPS problems with non-global zones

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

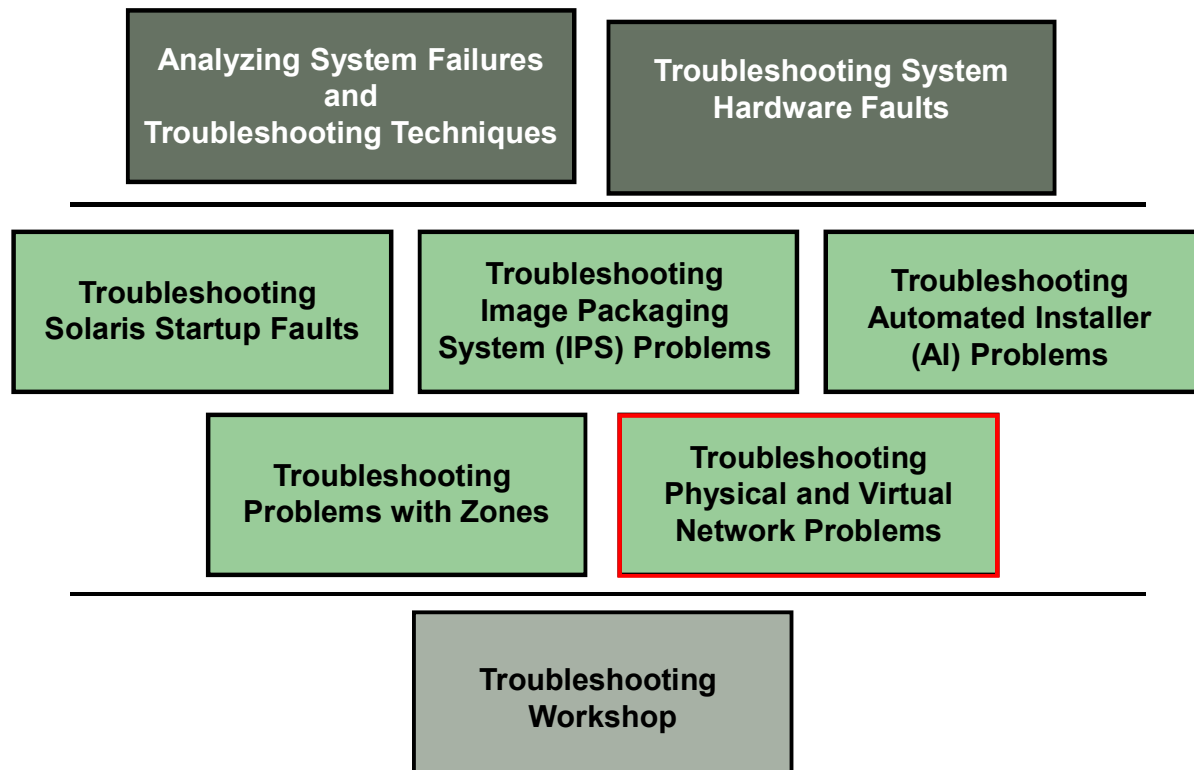
Troubleshooting Physical and Virtual Network Problems



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this lesson, Troubleshooting Physical and Virtual Network Problems, the focus is on potential issues with the network, their causes, and how to troubleshoot and resolve these issues. Given the size and complexity of the networking topic, it will not be possible to cover networking in great detail. To receive in-depth training on networking, it is recommended that you attend the Oracle Solaris 11 Network Administration course.

Objectives

After completing this lesson, you should be able to:

- Recall the physical network features and functionality
- Recall the virtual network features and functionality
- Verify the network configuration
- Display network information
- Observe network traffic on links
- Observe network traffic on flows
- Obtain detailed information about the network

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

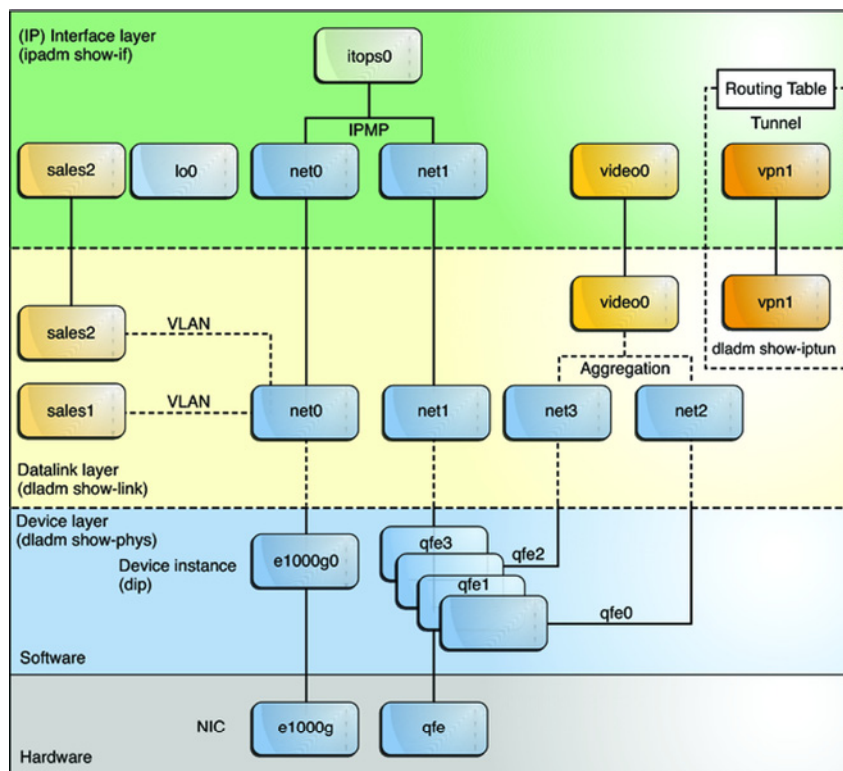
Agenda

- Reviewing Physical and Virtual Network Features and Functionality
- Checking the Basics
- Observing the Network
- Case Study 7

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 11 Network Implementation



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Oracle Solaris 11, the one-to-one relationship between the hardware, datalink, and interface layers remains. However, the software layer is decoupled from the hardware layer. With this separation, network configuration on the software level is no longer bound to the chipset or the network topology in the hardware layer. This implementation makes network administration more flexible in the following ways:

- The network configuration is insulated from any changes that might occur in the hardware layer. Link and interface configurations are preserved even if the underlying hardware is removed. These same configurations can then be reapplied to any replacement NIC, provided that the two NICs are of the same type.
- The separation of the network configuration from the network hardware configuration also allows the use of customized link names in the datalink layer.
- With the abstraction of the datalink layer, multiple networking abstractions or configurations such as VLANs, VNICs, physical devices, link aggregations, and IP tunnels are unified into a common administrative entity, which is the datalink.

This illustration provides an example of how administratively chosen names can be used in the network setup.

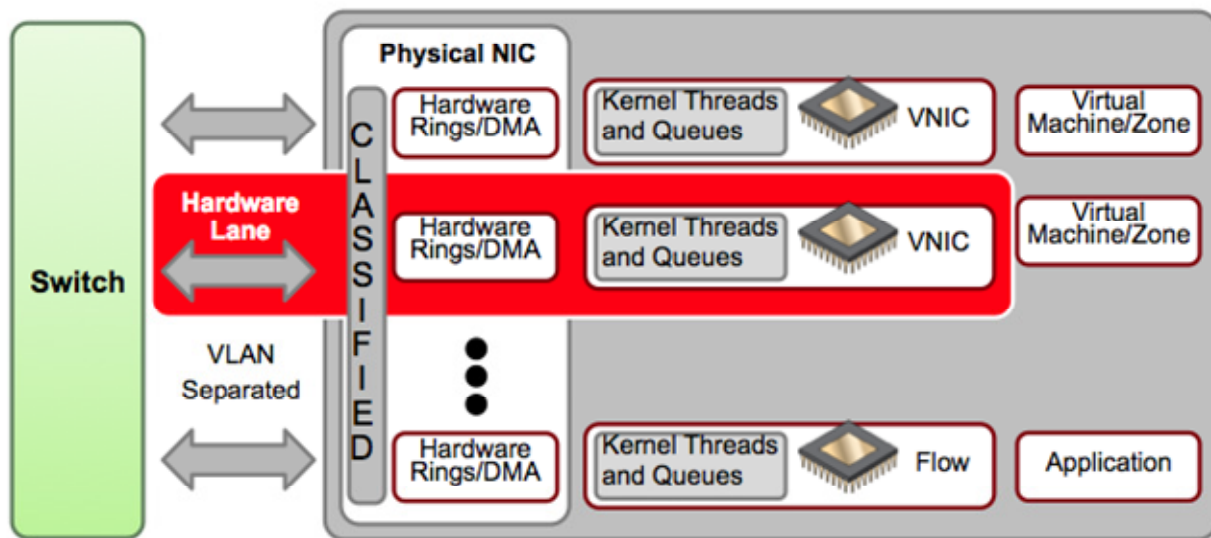
Virtual local area networks (VLANs) are configured on the net0 link. These VLANs, in turn, are also assigned customized names, such as sales1 and sales2. The VLAN sales2's IP interface is plumbed and operational.

- The datalinks of the device instances qfe0 and qfe2 are automatically named by the OS. These two datalinks are aggregated to host video feed. The aggregation's name can be customized. In the figure, the aggregation is named video0.
- Two interfaces (net0 and net1) with different underlying hardware (e1000g and qfe) are grouped together as an IP multipathing (IPMP) group called itops0.
- Two interfaces have no underlying devices: The tunnel vpn1 is configured for VPN connections, and lo0 exists for IP loopback operations.

All of the link and interface configurations in this figure are independent of the configurations in the underlying hardware. For example, if the qfe card is replaced, the video0 interface configuration for video traffic remains and can later be applied to a replacement NIC.

On the datalink layer of the same network stack implementation, you can configure bridges as shown in the following figure. Two interfaces, net0 and videoagg0, are configured as a bridge, bridge0. Packets that are received on one interface are forwarded to the other interface. After bridge configuration, both interfaces can still be used to configure VLANs and IP interfaces.

Virtual Network Architecture

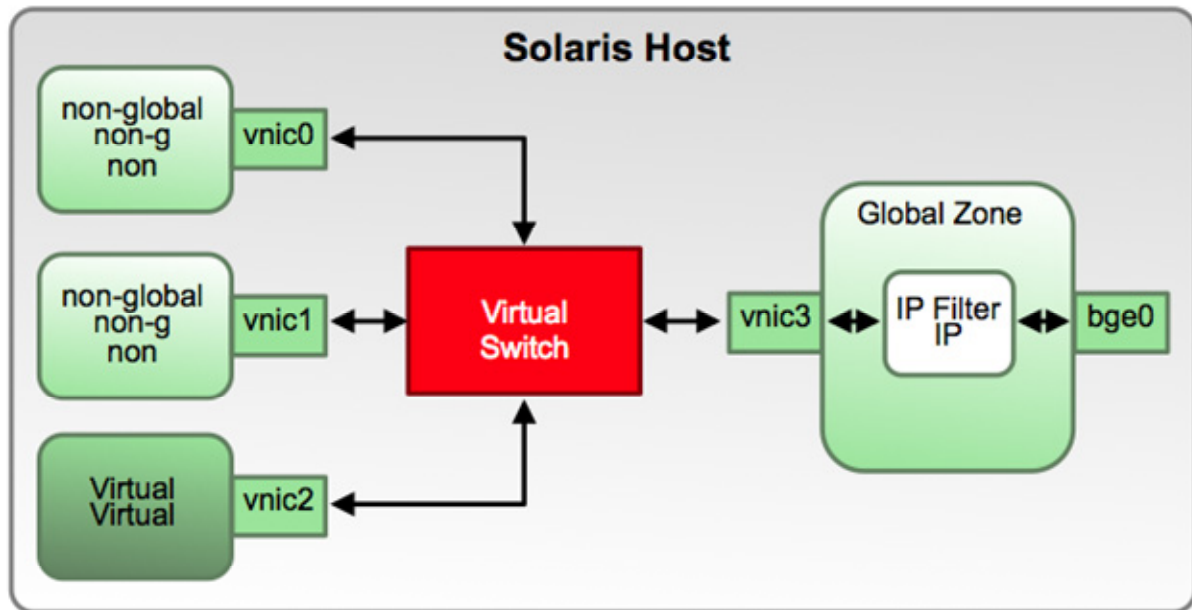


ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Review the main architecture of network virtualization and how a virtual network functions, as presented in the diagram in the slide. A network switch on the left side is shown with traffic going to a physical machine on the right side. There is the concept of hardware lanes that span from the hardware to the applications that allows for dedicated resources on a per-lane basis. The most common use case is to assign a hardware lane to a VNIC and then that VNIC to a zone. The networking processing for that zone is processed by the CPU and all the hardware resources for the zones are dedicated to processing for that zone, which has great performance advantages. Also note, the hardware lanes are extending to the left and that's where data center Ethernet comes in.

Virtual Network Topology



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Once VNICs are created on the physical NIC, the way they communicate with each other is through a virtual switch integrated within the stack. As shown in the graphic in the slide, an etherstub is created and acts like a Ethernet link, but it is virtual. VNICs are then created on top of the etherstub. The virtual switch is independent of any piece of hardware, which allows for the creation of some useful topologies.

Agenda

- Reviewing Physical and Virtual Network Features and Functionality
- **Checking the Basics**
- Observing the Network
- Case Study 7

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Observing and Troubleshooting IP Interfaces and Datalinks

- To observe and troubleshoot IP interface issues, use the `ipadm` and `ipadm show-*` commands.
- To observe and troubleshoot datalink issues, use the `dladm` and `dladm show-*` commands.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You should also check the IP interfaces with the `ipadm` and `ipadm show-*` commands and the datalinks with the `dladm` and `dladm show-*` commands. Both these commands and their subcommands and when to use them have been discussed in the next topic on observing the network.

Verifying the Status of the Network Services

Use `svcs | grep network` to verify network status.

```
# svcs | grep network
disabled Feb_08   svc:/network/nfs/server:default
online   Feb_08   svc:/network/tcp/congestion-control:vegas
...
maintenance      10:33:27 svc:/network/dns/server:default
...
```

Use `svcs -xv` to locate information concerning the failed service.

```
# svcs -xv network/dns/server
svc:/network/dns/server:default (BIND DNS server)
  State: maintenance since February 11, 2000 10:33:27 AM MST
  Reason: Start method exited with $SMF_EXIT_ERR_CONFIG.
    See: http://support.oracle.com/msg/SMF-8000-KS
    See: man -M /usr/man -s 1M named
    See: /var/svc/log/network-dns-server:default.log
  Impact: This service is not running.
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can check the status of the network services using the `svcs | grep network` command, as shown in the first example in the slide. You can also use the `svcs -xv` command to locate information concerning the failed (maintenance state) service.

Verifying That the Network Daemon is Running

To verify that the network daemon is running, use `ps -ef | grep inetd`.

```
# ps -ef | grep inetd
root  6034      1  0   Apr 21 ?           0:02 /usr/lib/inet/inetd start
root  2103      1  0   Apr 02 ?           0:29 /usr/lib/inet/inetd start
root  7261    7008  0 13:35:55 pts/1      0:00 grep inetd
root 16345      1  0   Apr 02 ?           0:00 /usr/lib/inet/inetd start
root 17144      1  0   Apr 02 ?           0:00 /usr/lib/inet/inetd start
#
```

To verify that the IPv6 daemon is running, use `ps -ef | grep in.ndpd`.

```
# ps -ef | grep in.ndpd
root  1827      1  0   Apr 02 ?           0:00 /usr/lib/inet/in.ndpd
root  7265    7008  0 13:38:07 pts/1      0:00 grep in.ndpd
#
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You might also want to check if the network daemon `inetd` is running by using the `ps -ef | grep inetd` command, as shown in the first example. If the `inetd` daemon is running, you see the output presented in the example.

If IPv6 is enabled on your network, verify that the IPv6 daemon `in.ndpd` is running by using the `ps -ef | grep in.ndpd` command. If the `in.ndpd` daemon is running, you see the output presented in the example.

Verifying the Router and Protocols

Task	Command
Display IPv4 or IPv6 forwarding and routing services.	<code>routeadm</code>
Display list of routes applied at start up.	<code>route show -p</code>
Flush the routing tables of all gateways.	<code>route -f</code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A router provides the interface between two or more networks. When a router is configured, a unique name and IP address are assigned to each of the router's physical network interfaces. Therefore, each router has a host name and an IP address that are associated with its primary network interface, in addition to a minimum of one more unique name and IP address for each additional network interface. You can configure two types of routing on an Oracle Solaris system: static and dynamic. You can configure either or both routing types on a single system. A system that implements dynamic routing relies on routing protocols, such as RIP for IPv4 networks, and RIPng for IPv6 networks, to route network traffic as well as to update routing information in the table. With static routing, routing information is maintained manually by the use of the `route` command. For complete details, refer to the `route(1M)` man page.

You can troubleshoot basic routing problems with the three commands presented in the slide.

A more detailed description of each command and how you can use the command for verification and troubleshooting is presented on the next page.

- **routeadm:** You can use this command to display routing configuration for the system, to include the current routing option (IPv4 or IPv6), the current routing configuration (enabled or disabled), and the system state. The routing services are also displayed ("route:default ripng:default").
- **route show -p:** The `route show` command can be used only in conjunction with the `-p` option.
Note: The `route` command operates on both IPv4 and IPv6 routes, with IPv4 routes as the default. If you use the `-inet6` option on the command line immediately after the `route` command, operations are performed on IPv6 routes.
- **route -f:** If you use the `-f` option in conjunction with any of the `route` subcommands, `route` flushes the gateways before performing the subcommand. You can specify which table to flush by placing the `inet` or `inet6` modifier immediately after the `-f` option. If unspecified, flushing IPv4 (`inet`) routes is the default.

Checking and Troubleshooting the DNS

1. Check the DNS service status by running `svcs -xv dns/server:default`.
2. Check the DNS service log file by running `tail /var/svc/log/network-dns-server:default.log`.
3. Check syslog messages by running `grep named /var/adm/messages`.
4. Start the `named` daemon manually by running `named -g`.
5. After the issue has been fixed, clear the maintenance required state by running `svcadm clear dns/server:default`.
6. Verify that the service is back online by running `svcs dns/server:default`.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To check the DNS service status and then troubleshoot the service if there is a problem, complete the steps presented in the slide.

Note for step 4: Running `named` in the foreground forces all logging to standard error so that it is easier to identify problems.

Checking and Troubleshooting the DNS Server

```
# svcs -xv network/dns/server
svc:/network/dns/server:default (BIND DNS server)
  State: maintenance since February 11, 2000 10:33:27 AM MST
Reason: Start method exited with $SMF_EXIT_ERR_CONFIG.
  See: http://support.oracle.com/msg/SMF-8000-KS
  See: man -M /usr/man -s 1M named
  See: /var/svc/log/network-dns-server:default.log
Impact: This service is not running.
# more /var/svc/log/network-dns-server:default.log
...
dns-server: Creating default rndc key file: /etc/rndc.key.
wrote key file "/etc/rndc.key"
dns-server : Configuration file /etc/named.conf does not exist!
[ Feb 11 17:33:27 Method "start" exited with status 96. ]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in the slide, the DNS server service is in a maintenance (failed) state. The `svcs -xv network/dns/server` command output directs you to view the `/var/svc/log/network-dns-server:default.log` file for more details on the error causing the DNS server to fail. The `network-dns-server:default.log` shows that the configuration file `/etc/named.conf` does not exist.

Checking and Troubleshooting NFS

This section covers the following topics:

- Checking connectivity on an NFS client
- Checking the NFS server remotely
- Verifying the NFS service on the server

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The primary points of possible failure with NFS are the client, the server, and the network. One approach to resolving NFS problems is by trying to isolate each individual component to find the one that is not working. In the slides that follow, you are shown how to check each primary point of failure.

Note: In all situations, the `mountd` and `nfsd` daemons must be running on the server for remote mounts to succeed.

Checking Connectivity on an NFS Client

1. Check that the NFS server is reachable from the client by running `/usr/sbin/ping NFS_server`.
2. If the server is not reachable from the client, ensure that the local name service is running.
3. If the name service is running, ensure that the client has received the correct host information by running `/usr/bin/getent hosts NFS_server`.
4. If the host information is correct, but the server is not reachable from the client, run the `ping` command from another client.
5. If the server is reachable from the second client, use `ping` to check connectivity of the first client to other systems on the local net.
6. If the software is correct, check the networking hardware.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To check the connectivity on an NFS client, complete the steps presented in the slide.

Note for step 1: If the command reports that the server is alive, remotely check the NFS server. See the next slide for instructions about how to complete this task.

Note for step 4: If the command run from a second client fails, see the slide on Verifying the NFS Service on the Server for instructions about how to complete this task.

Note for step 5: If this command fails, check the networking software configuration on the client, for example, `/etc/netmasks`, and the property information associated with the `svc:/system/name-service/switch` service.

Note for step 6: Try to move the client onto a second net drop.

Checking the NFS Server Remotely

1. Check that the NFS services have started on the NFS server by running `rpcinfo -s bee | egrep 'nfs | mountd'`.
2. Check that the server's `nfstd` processes are responding by running `/usr/bin/rpcinfo -u NFS_server nfs` on the client.
3. Check that the server's `mountd` is responding by running `/usr/bin/rpcinfo -u NFS_server mountd`.
4. Check if the local `autofs` service is being used by running `cd /net/wasp`.
5. Verify that file system is shared as expected on the server by running `/usr/sbin/showmount -e NFS_server`.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To check the NFS server remotely, complete the steps presented in the slide.

Note for step 1: If the daemons have not been started, start them by running the command `svcadm restart network/nfs/server`.

Note for step 2: This command tests the UDP NFS connections from the server. Note that NFS version 4 does not support UDP. If the server is running, it prints a list of program and version numbers. Using the `-t` option tests the TCP connection. If this command fails, complete the steps for Verifying the NFS Service on the Server presented on the next slide.

Note for step 3: If the server is running, it prints a list of program and version numbers. Using the `-t` option tests the TCP connection. If this command fails, complete the steps for Verifying the NFS Service on the Server presented on the next slide.

Note for step 4: Choose a `/net` or `/home` mount point that you know should work properly. If this command fails, then as `root` on the client, type `svcadm restart system/filesystem/autofs` to restart the `autofs` service.

Note for step 5: Check the entry on the server and the local mount entry for errors. Also, check the namespace. In this instance, if the first client is not in the `eng netgroup`, that client cannot mount the `/usr/src` file system. Check all entries that include mounting information in all the local files. The list includes `/etc/vfstab` and all the `/etc/auto_*` files.

Checking the NFS Server Remotely

1. Check that the NFS services have started on the NFS server by running `rpcinfo -s bee | egrep 'nfs | mountd'`.
2. Check that the server's `nfsd` processes are responding by running `/usr/bin/rpcinfo -u NFS_server nfs` on the client.
3. Check that the server's `mountd` is responding by running `/usr/bin/rpcinfo -u NFS_server mountd`.
4. Check if the local `autofs` service is being used by running `cd /net/wasp`.
5. Verify that file system is shared as expected on the server by running `/usr/sbin/showmount -e NFS_server`.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To check the NFS server remotely, complete the steps presented in the slide.

Note for step 1: If the daemons have not been started, start them by running the command `svcadm restart network/nfs/server`.

Note for step 2: This command tests the UDP NFS connections from the server. Note that NFS version 4 does not support UDP. If the server is running, it prints a list of program and version numbers. Using the `-t` option tests the TCP connection. If this command fails, complete the steps for Verifying the NFS Service on the Server presented on the next slide.

Note for step 3: If the server is running, it prints a list of program and version numbers. Using the `-t` option tests the TCP connection. If this command fails, complete the steps for Verifying the NFS Service on the Server presented on the next slide.

Note for step 4: Choose a `/net` or `/home` mount point that you know should work properly. If this command fails, then as `root` on the client, type `svcadm restart system/filesystem/autofs` to restart the `autofs` service.

Note for step 5: Check the entry on the server and the local mount entry for errors. Also, check the namespace. In this instance, if the first client is not in the `eng netgroup`, that client cannot mount the `/usr/src` file system. Check all entries that include mounting information in all the local files. The list includes `/etc/vfstab` and all the `/etc/auto_*` files.

Verifying the NFS Service on the Server

1. Check that the server can reach the clients by pinging a client.
2. If the client is not reachable from the server, ensure that the local name service is running.
3. If the name service is running, check the networking software configuration on the server, for example, `/etc/netmasks`, and the property information associated with the `svc:/system/name-service/switch` service.
4. Check whether the `rpcbind` daemon is running by typing `/usr/bin/rpcinfo -u localhost rpcbind`.
5. Check whether the `nfsd` daemon is running by typing `rpcinfo -u localhost nfs`.
6. Check whether the `mountd` daemon is running by typing `/usr/bin/rpcinfo -u localhost mountd`, followed by `ps -ef | grep mountd`.

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To verify the NFS service on the server, complete the steps presented in the slide.

Note for step 4: If the server is running, it prints a list of program and version numbers that are associated with the UDP protocol.

Note for step 5: If the server is running, it prints a list of program and version numbers that are associated with the UDP protocol. Use the `-t` option with `rpcinfo` to check the TCP connection. If these commands fail, restart the NFS service by running the command `svcadm restart network/nfs/server`.

Notes for step 6: If the server is running, it prints a list of program and version numbers that are associated with the UDP protocol. Use the `-t` option with `rpcinfo` to check the TCP connection. If these commands fail, restart the NFS service by running the command `svcadm restart network/nfs/server`.

Checking the Name-Service/Switch SMF Service

To check the name-service/switch SMF service configuration, use the following command:

```
# svccfg -s name-service/switch listprop config
config                                application
config/value_authorization astring    solaris.smf.value.name-
    service.switch
config/default                      astring    files
config/host                          astring    "files dns mdns"
config/printer                      astring    "user files"
...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The name-service/switch SMF service defines the search order of the network databases for configuration information. Some of the network configuration information that previously were stored in configuration files, such as the default domain, have been converted to become properties of this SMF service. The properties of this SMF service determines the implementation of the name services on the system.

The values that are set for each of the properties determine which name service to search for information that would affect network users, such as passwords, aliases, or network masks. In the example presented in the slide, the automount and password properties are set to `files` and `nis`. Thus, automount information and password information are obtained from files and from the NIS service.

Note: The name-service/switch SMF service has been modified to accommodate lookups using IPv6 addresses.

Quiz

You can use the `netadm show-events` command to monitor the current state of all network connections.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Which command can you use to display IPv4 or IPv6 forwarding and routing services?

- a. route monitor
- b. routeadm
- c. route show
- d. routeadm -r ipv4-forwarding

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Agenda

- Reviewing Physical and Virtual Network Features and Functionality
- Checking the Basics
- **Observing the Network**
- Case Study 7

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Using `netstat` to Display Network Information

Task	Command
Display the status of transport protocols.	<code>netstat</code>
Display the protocol statistics.	<code>netstat -s</code>
Display the network interface status.	<code>netstat -i</code>
Display the sockets status.	<code>netstat -a</code>
Display the status of transmissions for packets of a specific address type.	<code>netstat -f <i>inet</i> / <i>inet6</i></code>
Display the status of known routes.	<code>netstat -r</code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `netstat` command generates displays that show network status and protocol statistics. You can display the status of TCP, SCTP, and UDP endpoints in table format. You can also display routing table information and interface information.

Note: The `netstat` command displays both IPv4 and IPv6 network status. You can choose which protocol information to display by setting the `DEFAULT_IP` value in the `/etc/default/inet_type` file or by using the `-f` command-line option. With a permanent setting of `DEFAULT_IP`, you can ensure that `netstat` displays only IPv4 information. You can override this setting by using the `-f` option.

The table in the slide presents a number of network monitoring tasks and the `netstat` command used to complete the task. A description of what each command does follows:

- **`netstat`:** Displays the status of the transport protocols. To display the status of a particular transport protocol on a system, use the `-P` option with the transport-protocol variable: `tcp`, `sctp`, or `udp`.
- **`netstat -s`:** Displays protocol statistics for the UDP, TCP, SCTP, ICMP, and IP protocols. The output can indicate areas where a protocol is having problems.
- **`netstat -i`:** Shows the state of the network interfaces that are configured on the local system. With the `-i` option, you can determine the number of packets a system transmits and receives on each network.

- **netstat -a:** Displays the status of sockets and routing table entries.
- **netstat -f *inet* / *inet6*:** Displays statistics related to packet transmissions of a particular address family. Use *inet* as the argument to **netstat -f** to view IPv4 transmission information and *inet6* to view IPv6 information.
- **netstat -r:** Displays the routing table for the local host. The routing table shows the status of all routes that the host knows about.

Using netstat to Display Network Information: Example

```
# netstat -r
```

Routing Table: IPv4					
Destination	Gateway	Flags	Ref	Use	Interface
-					
default	session	UG	1	0	net0
host02-local	host02-local	UH	2	796	lo0
192.168.124.0	host02	U	5	301229	net0

Routing Table: IPv6					
Destination/Mask	Gateway	Flags	Ref	Use	If
host02-loc	host02-local	UH	2	228	lo0
fe80::/10	fe80::214:4fff:fe02:5cf6	U	2	0	net0

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in the slide, the `netstat` command is used to display routing information.

Using `ipadm` to Display IP Interface and Address Information

Task	Command
Obtain general information about IP interfaces.	<code>ipadm</code>
Obtain interface information only.	<code>ipadm show-if</code>
Obtain address information only.	<code>ipadm show-addr</code>
Obtain information about interface properties.	<code>ipadm show-ifprop</code>
Obtain information about address properties.	<code>ipadm show-addrprop</code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use the `ipadm` command to monitor and obtain information about IP interfaces and their properties. By itself, the command displays general information about IP interfaces on the system. However, you can also use the `ipadm show-*` commands to restrict the information that you want to display.

The table in the slide presents a number of IP interface and address monitoring tasks and the `ipadm` command used to complete the task. A description of what each command does follows:

- **`ipadm`:** Provides default information about all the system's IP interfaces. The information includes the name, class, and state of each IP interface as well as the actual address of each interface.
- **`ipadm show-if`:** Displays information about IP interfaces. If you do not specify an interface, the information covers all the interfaces on the system.
- **`ipadm show-addr`:** Displays information about IP addresses. If you do not specify an interface, information about all the IP addresses on the system is provided.
- **`ipadm show-ifprop`:** Displays information about properties of IP interfaces. If you do not specify a property or an interface, information about all the properties of all the IP interfaces on the system is provided.

- **ipadm show-addprop:** Displays information about IP address properties. To list all the properties, omit the *addrobj* option. To list a single property for all the IP addresses, specify only the property. To list all the properties of a specific address, specify only the *addrobj* option.

Displaying Network Information: Example

```
# ipadm show-if
IFNAME      CLASS      STATE      ACTIVE OVER
lo0          loopback  ok         yes   --
net0         ip        ok         yes   --
root@host02:~# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4        static    ok         127.0.0.1/8
lo0/zoneadmd.v4 static    ok         127.0.0.1/8
lo0/zoneadmd.v4a static    ok         127.0.0.1/8
lo0/zoneadmd.v4b static    ok         127.0.0.1/8
net0/v4        static    ok         192.168.124.21/24
net0/zoneadmd.v4 static    ok         192.168.124.24/24
net0/zoneadmd.v4a static    ok         192.168.124.25/24
net0/zoneadmd.v4b static    ok         192.168.124.26/24
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in the slide, the `dladm show-phys` command shows that there are four Ethernet network interfaces configured in this system. The `ipadm show-if` is used to display the network interfaces that are currently configured. The `ipadm show-addr` command shows the current IP address configuration.

Using `dladm` to Display Datalink Information

Task	Command
Display general information about datalinks.	<code>dladm</code>
Display a system's datalinks.	<code>dladm show-link</code>
Display physical attributes of datalinks	<code>dladm show-phys</code>
Displays information about the VNICs on the system.	<code>dladm show-vnic</code>
Displays information about the datalink properties.	<code>dladm show-linkprop</code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use the `dladm show` subcommands to display detailed datalink configuration information. By itself, the command displays general information about datalinks on the system. However, you can also use the `dladm show-*` commands to restrict the information that you want to display.

The table in the slide presents a number of datalink monitoring tasks and the `dladm` command used to complete the task. A description of what each command does follows:

- **`dladm`:** Displays general information about the system's datalinks, including their class, state, and underlying physical links.
- **`dladm show-link`:** Displays the datalinks on a system. A system has as many datalinks as installed NICs. You can use options with this command to customize the information you obtain. For example, using the `-P` option includes persistent configuration information about the datalinks. Based on the information provided by this command, you can proceed with further investigation into the network configuration.

- **dladm show-phys:** Displays information about the system's datalinks in relation to the physical NICs with which they are associated. The output shows, among other details, the physical NICs with which the datalinks with generic link names are associated. Another useful option for the command is `-L`, which shows the physical location for each datalink. The location determines the instance number of the datalink such as `net0`, `net1`, and so on.
- **dladm show-vnic:** Displays information about the VNICs on the system. Because VNICs are also datalinks, any `dladm` command that shows information about datalinks includes information about VNICs if these exist on the system. For example, `dladm show-link` includes VNICs on the list. Or, you can use the `dladm show-linkprop` command to check the properties of VNICs. To obtain property information about a single VNIC, specify the VNIC when you display link properties.

Displaying Datalink Information: Example

```
## dladm show-phys
LINK          MEDIA          STATE      SPEED  DUPLEX    DEVICE
net1          Ethernet        unknown    0       unknown   e1000g1
net0          Ethernet        up         100     full      e1000g0
net2          Ethernet        unknown    0       unknown   e1000g2
net3          Ethernet        unknown    0       unknown   e1000g3
root@host02:~# dladm show-link
LINK          CLASS      MTU      STATE  OVER
net1          phys      1500     unknown --
net0          phys      1500     up     --
net2          phys      1500     unknown --
net3          phys      1500     unknown --
etherstub0    etherstub 9000     unknown --
vnic0         vnic      9000     up     etherstub0
vnic1         vnic      9000     up     etherstub0
...
# dladm show-linkprop net0
LINK          PROPERTY  PERM  VALUE      DEFAULT    POSSIBLE
net0          speed     r-    100        100        --
net0          autopush  rw    --         --         --
net0          zone     rw    --         --         --
net0          duplex   r-    full       full       half,full
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Using `dlstat` to Observe Network Traffic on Links

Task	Command
Display inbound and outbound traffic statistics per lane	<code>dlstat [link]</code> <code>dlstat show-link [link]</code>
Display inbound and outbound traffic statistics per network physical device	<code>dlstat show-phys [link]</code>
Display inbound and outbound traffic statistics per port per aggregation	<code>dlstat show-aggr [link]</code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can use the `dlstat` command to monitor and obtain statistics about network traffic on datalinks.

The table in the slide presents a number of network traffic monitoring tasks and the `dlstat` command used to complete the task. A description of what each command does follows:

- **`dlstat` and `dlstat show-link`:** Provide statistics that refer to the lanes that are configured over the physical link. The lanes are constituted by the datalinks.
- **`dlstat show-phys`:** Provides statistics that refer to the physical network device. The output shows both incoming and outgoing traffic statistics on each link on the system. The number of packets and their byte sizes are displayed.
- **`dlstat show-aggr`:** Shows network packet statistics for each aggregation's ports when traffic traverses the aggregation on the system. As network traffic is received or sent by the system through the aggregation, information about incoming and outgoing packets and their respective sizes is reported for every port. The ports are identified by the underlying links of the aggregation.

Note: You can use either the `-r` option or the `-t` option to the `dlstat` command to restrict the statistics information to the receive side or the transmit side, respectively.

Using `dlstat` to Observe Network Traffic on Links: Example

```
# dlstat show-phys
```

LINK	IPKTS	RBYTES	OPKTS	OBYTES
net1	0	0	0	0
net2	0	0	0	0
net0	4.99M	2.30G	2.45M	666.46M
net3	0	0	0	0

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in the, the `dlstat show-phys` command shows that there is only one active network interface on this system.

Using `flowstat` to Observe Network Traffic on Flows

The `flowstat` command options:

- **`[-r | -t]`**: Displays either receive-side statistics only (`-r` option) or transmit-side statistics only (`-t` option)
- **`-i interval`**: Specifies the time in seconds at which you want the displayed statistics to be refreshed
- **`-l link`**: Indicates that you want to monitor the statistics for all the flows on the specified datalink
- **`flow`**: Indicates that you want to monitor the statistics of a specified flow only

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flow statistics help you evaluate packet traffic on any defined flows on the system. To obtain flow information, you use the `flowstat` command with the following options:

- **`[-r | -t]`**: Displays either receive-side statistics only (`-r` option) or transmit-side statistics only (`-t` option). To display statistics for both the receive-side and the transmit-side, omit the option.
- **`-i interval`**: Specifies the time in seconds at which you want the displayed statistics to be refreshed. If you do not use this option, then static output is displayed.
- **`-l link`**: Indicates that you want to monitor the statistics for all the flows on the specified datalink. If you do not use this option, then information about all the flows on all the datalinks is displayed.
- **`flow`**: Indicates that you want to monitor the statistics of a specified flow only. If you do not use this option, then depending on whether you specified a link, all flow statistics are displayed.

Using `flowadm` to Display Flow Information

Task	Command
Display information about flows that have been created on the system.	<code>flowadm</code> <code>flowadm show-flow</code>
Display detailed property information for flows	<code>flowadm flow-prop [flow]</code>

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flow control can be an important tool in the overall network performance strategy. But, if not used wisely, flow controls can cause performance issues for end-users. You can use the `flowadm` command to display detailed flow configuration information. By itself, the command displays general information about flows on the system. However, you can also use the `flowadm` with subcommands to restrict the information that you want to display.

The table in the slide presents a few of flow monitoring tasks and the `flowadm` command used to complete the task. A description of what each command does follows:

- **`flowadm` and `flowadm show-flow`:** Displays general information about the system's flows that have been created over a link
- **`flowadm flow-prop`:** Displays detailed property information for flows

Configure and Display Flow Information: Example

```
# flowadm add-flow -l e000g0 -a transport=TCP,local_port=443 flow1
# flowadm show-flow -l e1000g0
FLOW      LINK      IP ADDR      PROTO  PORT  RPORT  DSFLD
flow1     e1000g0  --           tcp    443   --     --
# flowadm set-flowprop -p maxbw=500M flow1
# flowadm show-flowprop flow1
FLOW      PROPERTY  VALUE      DEFAULT  POSSIBLE
flow1     maxbw     500        --        --
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in the slide, the `flowadm add-flow` command defines a flow named `flow1` for TCP port 443 on network interface `e1000g0`. The `flowadm set-flowprop` command limits the maximum bandwidth of `flow1` to 500 MB.

Using `flowstat` to Observe Network Traffic on Flows: Example

```
# flowstat -i 1
FLOW      IPKTS      RBYTES  IERRS      OPKTS      OBYTES      OERRS
flow1 727.85K      1.73G      0    277.34K      15.32M      0
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the example presented in the slide, the `flowstat` command shows information every second about incoming and outgoing traffic on all configured flows on the system.

Quiz

Which commands can you use to display inbound and outbound traffic statistics per lane? (Select two.)

- a. `dlstat [link]`
- b. `dlstat show-phys [link]`
- c. `dlstat show-link [link]`
- d. `dlstat show-aggr [link]`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a, c

Quiz

Which command should you use if you want to obtain only IP interface information?

- a. `ipadm show-addr`
- b. `ipadm show-addrprop`
- c. `ipadm show-if`
- d. `ipadm show-ifprop`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: c

Agenda

- Reviewing Physical and Virtual Network Features and Functionality
- Checking the Basics
- Observing the Network
- Case Study 7

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Can't Access Network!

- Clarify problem statement.
 - System `host2` cannot access the network after a scheduled software update event!
- Using “assumptive” method, prioritize the most probable causes of failure.
 - Problem with updated software
 - Configuration error
 - Hardware fault
- Starting with the most probable cause of failure, gather pertinent information.
 - Peruse system message log.
 - Run system diagnostics.
 - Check software package

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Troubleshooting Steps

Perform the following steps to identify the cause of the problem:

1. Verify that system cannot access network.
2. Search `/var/adm/messages` file for error messages.
3. Query fault manager for faulty components.
4. List the probable cause (s) of the problem.
5. Verify configuration.
6. Verify hardware.
7. Verify associated software packages.
8. Repair the fault.
9. Verify the solution.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the high-level steps involved in this case study.

Case Study 7: Walkthrough

```

Sun Fire T200, No Keyboard
Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.30.4.d, 16256 MB memory available, Serial #67263734.
Ethernet address 0:14:4f:2:5c:f6, Host ID: 84025cf6.
Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0,0:a File and args:
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Hostname: host02

host02 console login: root
Password: XXXXXX
Mar 19 22:12:03 host02 login: ROOT LOGIN /dev/console
Last login: Tue Mar 19 22:04:20 on console
Oracle Corporation      SunOS 5.11      11.1      April 2013
root@host02:~#
Mar 19 22:12:18 host02 zoneadmd[2757]: [zone 'database'] WARNING: skipping network
        interface 'net0' which may not be present/plumbed in the global zone.: No such
        device or address
Mar 19 22:12:18 host02 zoneadmd[2756]: [zone 'storage'] WARNING: skipping network
        interface 'net0' which may not be present/plumbed in the global zone.: No such
        device or address
Mar 19 22:12:18 host02 zoneadmd[2758]: [zone 'web'] WARNING: skipping network
        interface 'net0' which may not be present/plumbed in the global zone.: No such
        device or address
root@host02:~#
##### Note the warning messages during system startup. #####

```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide begin the Case Study 4 walkthrough. Note that line beginning with “#####” are comments.

Case Study 7: Walkthrough

```
##### Test network functionality. #####
root@host02:~# ping 192.168.124.10
ping: sendto No route to host
##### Query fault manager for faulty components. #####
root@host02:~# fmadm faulty
root@host02:~#
##### Check /var/adm/messages file for errors. #####
root@host02:~# cat /var/adm/messages | grep error
root@host02:~#
##### Verify network configuration. #####
root@host02:~# ipadm show-if
IFNAME      CLASS      STATE      ACTIVE OVER
lo0         loopback  ok         yes    --
net0        ip        disabled  no     --
root@host02:~# dladm show-link
root@host02:~#
root@host02:~# dladm show-phys
root@host02:~#
##### Note that there appears to be no network interfaces in this system. #####
##### Bring system to OpenBoot PROM (OBP). #####
root@host02:~# init 0
...
ok
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Walkthrough

```
##### Run firmware diagnostics. #####
ok setenv diag-switch? true
diag-switch? =          true
ok setenv auto-boot? false
auto-boot? =            false
ok ok reset-all
SC Alert: Host System has Reset
...
Device: pci
/pci@780: Device 0 pci
/pci@780/pci@0: Device 1 pci
/pci@780/pci@0/pci@1: Device 0 network network
/pci@7c0/pci@0: Device 2 pci
/pci@7c0/pci@0/pci@2: Device 0 network network
...
##### No network errors are indicated. #####
##### Boot the system. #####
##### Run firmware diagnostics. #####
ok setenv diag-switch? false
diag-switch? =          false
ok setenv auto-boot? true
auto-boot? =            true
ok boot
...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Walkthrough

```

host02 console login: root
Password: XXXXXX
Mar 19 22:12:03 host02 login: ROOT LOGIN /dev/console
Last login: Tue Mar 19 22:04:20 on console
Oracle Corporation      SunOS 5.11      11.1      April 2013

root@host02:~#
Mar 19 22:12:18 host02 zoneadmd[2757]: [zone 'database'] WARNING: skipping network
        interface 'net0' which may not be present/plumbed in the global zone.: No such
        device or address
Mar 19 22:12:18 host02 zoneadmd[2756]: [zone 'storage'] WARNING: skipping network
        interface 'net0' which may not be present/plumbed in the global zone.: No such
        device or address
Mar 19 22:12:18 host02 zoneadmd[2758]: [zone 'web'] WARNING: skipping network
        interface 'net0' which may not be present/plumbed in the global zone.: No such
        device or address
##### Run the prtdiag utility and check for network errors. #####
root@host02:~# prtdiag |grep network
IOBD/NET-1    PCIE IOBD      /pci@780/pci@0/pci@1/network    network-pciex8086,105e
IOBD/NET-1    PCIE IOBD      /pci@780/pci@0/pci@1/network    network-pciex8086,105e
IOBD/NET-1    PCIE IOBD      /pci@7c0/pci@0/pci@2/network    network-pciex8086,105e
IOBD/NET-1    PCIE IOBD      /pci@7c0/pci@0/pci@2/network    network-pciex8086,105e
##### Network interfaces appear to be OK. #####

```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Walkthrough

```
##### Verify that the network interface have been enumerated in the #####
##### path_to_inst file . #####
root@host02:~# cat /etc/path_to_inst | grep network
"/pci@780/pci@0/pci@1/network@0" 0 "bge"
"/pci@780/pci@0/pci@1/network@0,1" 1 "bge"
##### Note that the network instance name is bge. #####
##### Using the instance name, determine the payload of the network #####
##### device driver package. #####
root@host02:~# pkg contents bge
PATH
kernel
kernel/drv
kernel/drv/bge.conf
kernel/drv/sparcv9
kernel/drv/sparcv9/bge
usr/share/man/man7d
usr/share/man/man7d/bge.7d
##### Verify the bge device driver package. #####
root@host02:~# pkg verify -v bge
PACKAGE                                STATUS
pkg://solaris/driver/network/ethernet/bge ERROR
      file: kernel/drv/sparcv9/bge
      Missing: regular file does not exist
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Walkthrough

```
##### Repair the bge device driver. #####
root@host02:~# pkg fix bge
Verifying: pkg://solaris/driver/network/ethernet/bge          ERROR
file: kernel/drv/sparcv9/bge
      Missing: regular file does not exist
Created ZFS snapshot: 2013-03-20-04:50:01
Repairing: pkg://solaris/driver/network/ethernet/bge
Creating Plan (Evaluating mediators): \pkg: Requested "fix" operation would affect
      files that cannot be modified in live image.
Please retry this operation on an alternate boot environment.
root@host02:~# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      NR      /           9.85G  static 2013-03-19 19:07
solaris.orig -      -           1.12M  static 2013-03-19 21:28
root@host02:~# beadm activate solaris.orig
root@host02:~# reboot
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Walkthrough

```

host02 console login: root
Password: XXXXXX
Mar 19 23:32:03 host02 login: ROOT LOGIN /dev/console
Last login: Tue Mar 19 23:04:20 on console
Oracle Corporation      SunOS 5.11      11.1      April 2013

root@host02:~# beadm mount solaris /mnt
root@host02:~# pkg -R /mnt fix bge
Verifying: pkg://solaris/driver/network/ethernet/bge          ERROR
file: kernel/drv/sparcv9/bge
Missing: regular file does not exist
Created ZFS snapshot: 2013-03-20-05:04:11
Repairing: pkg://solaris/driver/network/ethernet/bge
Creating Plan (Evaluating mediators): \

DOWNLOAD                                PKGS          FILES      XFER (MB)   SPEED
Completed                              1/1           1/1        0.1/0.1     0B/s

PHASE                                ITEMS
Updating modified actions              1/1
Updating image state                   Done
Creating fast lookup database          Done
root@host02:~# beadm unmount solaris
root@host02:~# beadm activate solaris
root@host02:~# reboot
...

```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Case Study 7: Walkthrough

```
##### Verify network repair. #####
host02 console login: root
Password: XXXXXX
Mar 19 23:13:40 host02 login: ROOT LOGIN /dev/console
Last login: Tue Mar 19 22:36:21 on console
Oracle Corporation      SunOS 5.11      11.1      April 2013
Welcome to D80397GC10_Lab on host02
root@host02:~# pkg verify bge
root@host02:~#
root@host02:~# dladm show-phys
LINK           MEDIA           STATE          SPEED  DUPLEX  DEVICE
net1           Ethernet        unknown        0      unknown bge1
net0           Ethernet        up             100    full    bge0
root@host02:~# dladm show-link
LINK           CLASS    MTU    STATE  OVER
net1           phys     1500   unknown --
net0           phys     1500   up     --
root@host02:~# ping 192.168.124.1
192.168.124.1 is alive
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Additional Resources

- OTN Oracle Solaris 11.1 Virtualization Technology:
<http://www.oracle.com/technetwork/server-storage/solaris11/technologies/virtualization-306056.html>
- OTN Oracle Solaris 11.1 Networking Virtualization Technology Page (white papers, blogs, etc.):
<http://www.oracle.com/technetwork/server-storage/solaris11/technologies/networkvirtualization-312278.html>
- Documentation:
 - Configuring and Administering Oracle Solaris 11.1 Networks
http://docs.oracle.com/cd/E26502_01/html/E28989/index.html
 - Using Virtual Networks in Oracle Solaris 11.1
http://docs.oracle.com/cd/E26502_01/html/E28992/index.html

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 8: Overview

- 8-1: Troubleshooting a network performance problem
- 8-2: Troubleshooting an inaccessible host problem
- 8-3: Troubleshooting a virtual network problem

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Recall the physical network features and functionality
- Recall the virtual network features and functionality
- Verify the network configuration
- Display network information
- Observe network traffic on links
- Observe network traffic on flows
- Obtain detailed information about the network

ORACLE

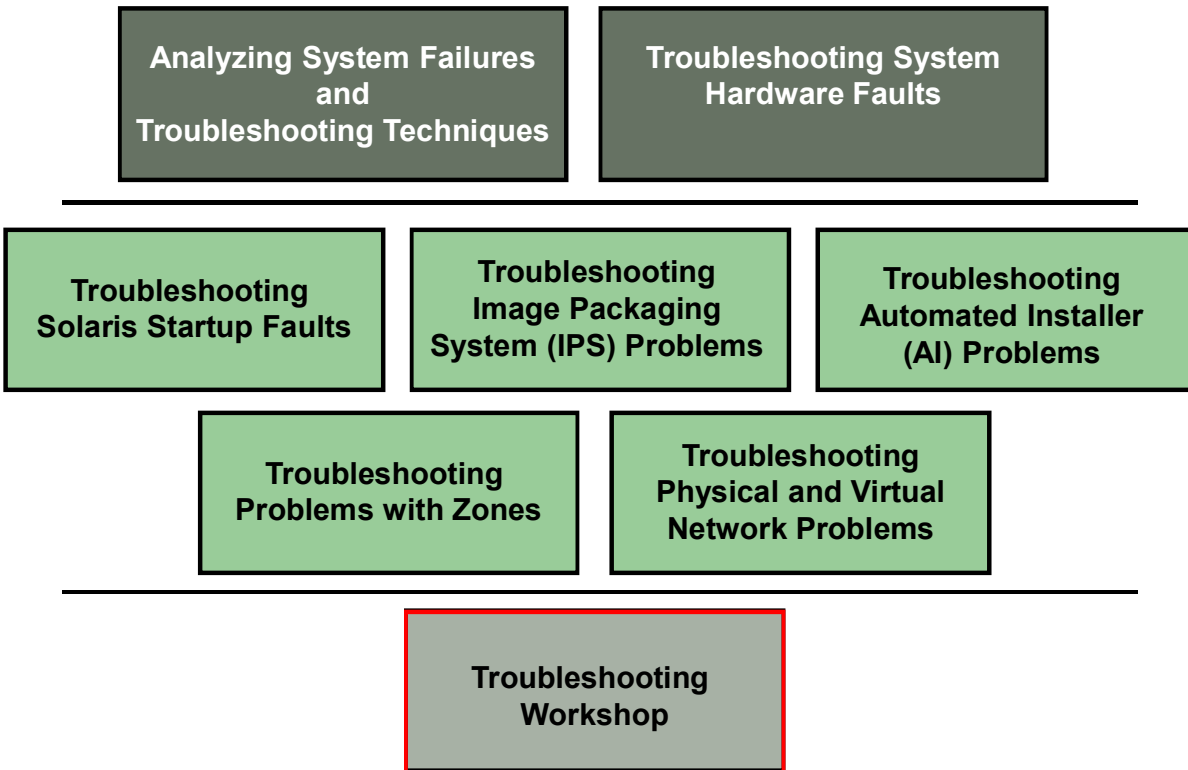
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Troubleshooting Workshop

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Troubleshoot SPARC-based system boot problems
- Troubleshoot system hang problems
- Troubleshoot Zones problems
- Troubleshoot SMF services problems
- Troubleshoot network problems
- Analyze a system crash dump
- Troubleshoot using DTrace and the Modular Debugger

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice

This practice covers the following topics:

- User and root logins
- Home directories
- Booting and system hangs
- Package installations
- Booting zones
- Kernel and system panics
- Commands not working
- Network connectivity
- Slowness in throughput

Note: This practice contains no step-by-step guidance. You are encouraged to use previous lesson materials as reference.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Unlike previous practices, this practice contains no step-by-step guidance. This is designed to be a two-day workshop. See the Activity Guide for more instructions and preparation.

Summary

In this lesson, you should have learned how to:

- Troubleshoot SPARC-based system boot problems
- Troubleshoot system hang problems
- Troubleshoot Zones problems
- Troubleshoot SMF services problems
- Troubleshoot network problems
- Analyze a system crash dump
- Troubleshoot using DTrace and the Modular Debugger

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Integrated Lights-Out Manager (ILOM)



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Introducing Oracle ILOM
- Configuring host server management actions
- Collecting system information and monitoring health status
- Setting up alert notifications and Syslog server for event logging
- Managing event and audit log entries
- Observing and debugging system behavior
- Managing Sun hardware faults through the Oracle ILOM Fault Management Shell

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This appendix provides an overview of the Oracle Integrated Lights Out Manager (ILOM) features and functions and a preview of the system troubleshooting tasks you can perform on a SPARC system by using Oracle ILOM.

The information provided in this appendix is not intended to be a comprehensive guide about how to use Oracle ILOM. For more complete information about the product, see the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Documentation Collection*.

Note: To locate Oracle ILOM 3.1 documentation that is specific to your Sun server platform, see the Oracle ILOM section of the administration guide that is available for your server.

Introducing Oracle ILOM

This section covers the following topics:

- Oracle ILOM: Overview
- Identifying key Oracle ILOM features and their functionality
- Identifying the supported management interfaces
- Getting started

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle ILOM: Overview

Oracle ILOM:

- Enables you to manage and monitor your Sun hardware
- Is pre-installed on all Sun rackmount servers, Blade servers, and chassis monitoring modules (CMMs)
- Has a service processor (SP) that runs its own embedded operating system and has a dedicated Ethernet port
- Automatically initializes as soon as power is applied to the Sun server
- Provides a number of interface options, to include browser-based web, CLI, SNMP, and IPMI

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Integrated Lights Out Manager (ILOM) provides advanced service processor (SP) hardware and software that you can use to manage and monitor your Sun hardware. Oracle ILOM is pre-installed on all Sun rackmount servers, blade servers, and chassis monitoring modules (CMMs).

The Oracle ILOM service processor (SP) runs its own embedded operating system and has a dedicated Ethernet port, which together provide out-of-band management capability. Oracle ILOM automatically initializes as soon as power is applied to the Sun server. It provides a full-featured, browser-based web interface and has an equivalent command-line interface (CLI). There is also an industry-standard Simple Network Management Protocol (SNMP) interface and Intelligent Platform Management Interface (IPMI) interface.

Identifying Key Oracle ILOM Features and Their Functionality

Oracle ILOM Feature	What You Can Do
Newly designed web and command-line interfaces	Display high-level information in a simple, standardized format that is common across x86 SP, SPARC SP, and CMM platforms.
Dedicated service processor and resources	<ul style="list-style-type: none"> • Manage the server without consuming system resources. • Continue to manage the server by using standby power even when the server is powered off.
Simple Oracle ILOM initial configuration	<ul style="list-style-type: none"> • Oracle ILOM automatically learns the network address of the server SP or CMM using IPv4 and IPv6 default settings. • Configure BIOS settings on the x86 SP platform.
Downloadable firmware updates	Download firmware updates using the browser-based web interface.
Remote hardware monitoring	<ul style="list-style-type: none"> • Monitor system health and system event logs. • Monitor hardware event logs. • Monitor audit event logs. • Monitor customer-replaceable units (CRUs) and field-replaceable units (FRUs), including power supplies, fans, host bus adapters (HBAs), PCI devices, disks, CPUs, memory, and motherboard. • Monitor environmental temperatures (component temperatures).



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle ILOM offers a full set of features, functions, and protocols that help you monitor, manage, and troubleshoot your server systems. The tables in this slide and the next present key Oracle ILOM features and their associated functionality.

Note for firmware updates: Updates to the Oracle ILOM 3.1 firmware are available through stand-alone software updates that you can download from the My Oracle Support (MOS) website for each Sun server or Sun Blade chassis system.

Identifying Key Oracle ILOM Features and Their Functionality

Oracle ILOM Feature	What You Can Do
Hardware and FRU inventory and presence	<ul style="list-style-type: none">• Identify installed CRUs and FRUs and their status.• Identify part numbers, versions, and product serial numbers.• Identify NIC card MAC addresses.
Remote KVMs	<ul style="list-style-type: none">• Redirect the system serial console through serial port and LAN.• Access keyboard, video, and mouse (KVM) on remote x86 systems and on some SPARC systems.• Redirect the OS graphical console to a remote client browser.• Connect a remote CD/DVD/floppy to the system for remote storage.
System power control and monitoring	<ul style="list-style-type: none">• Power the system on or off, either locally or remotely.• Force power-off for immediate shutdown or perform a graceful shutdown to shut down the host operating system before power-off.• Monitor power management and power history charts through the web interface.

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Identifying Key Oracle ILOM Features and Their Functionality

Oracle ILOM Feature	What You Can Do
Error and fault management	<ul style="list-style-type: none"> • Log events in a consistent method for all “service” data. • Monitor hardware and system-related errors, as well as ECC memory errors, reported on a dedicated user interface page, and into SP logs, syslog, and remote log host. • Oracle ILOM automatically clears most fault conditions after you perform a service action to address the fault.
System alerts, including SNMP traps, IPMI PETs, remote syslog, and email alerts	Monitor components by using industry-standard SNMP commands and the IPMItool utility.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Identifying the Supported Management Interfaces

- **Web interface:** Enables you to access the Oracle ILOM SP or CMM through a web browser
- **Command-line interface (CLI):** Enables you to access the Oracle ILOM CLI on the server SP or CMM by using an SSH client
- **Intelligent Platform Management Interface (IPMI):** Enables the management of server systems over a number of different types of networks
- **WS-Management/CIM:** Enables developers to build and deploy network management applications to monitor and manage information about Oracle's Sun system hardware
- **Simple Network Management Protocol (SNMP) interface:** Provides an SNMP v3.0 interface for third-party applications, such as HP OpenView and IBM Tivoli

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can access all of the Oracle ILOM features and functions by using any of, or a combination of all, the interfaces and protocols that Oracle ILOM supports. The supported management interfaces are listed in the slide. Additional information on each interface follows:

- **Web interface:** From the Oracle ILOM web interface, you can perform daily system management operations remotely. Additionally, from the web interface, you can launch tools to redirect KVMS, or to perform maintenance and diagnostic operations.
- **Command-line interface (CLI):** The command-line interface enables you to perform server management operations remotely by using industry-standard DMTF-style keyboard commands and scripting protocols.
- **Intelligent Platform Management Interface (IPMI):** IPMI is an open industry-standard interface. IPMI functionality includes field-replaceable unit (FRU) inventory reporting, system monitoring, logging of system events, system recovery (including system resets and power-on and power-off capabilities), and alerting.
- **WS-Management/CIM:** As of version 3.0.8, Oracle ILOM supports the use of the Distributed Management Task Force (DMTF) Web Services for Management (WS-Management) protocol and Common Information Model (CIM).

- **Simple Network Management Protocol (SNMP) interface:** Some of the MIBs supported by Oracle ILOM include:
 - SUN-PLATFORM-MIB
 - SUN-ILOM-CONTROL-MIB
 - SUN-HW-TRAP-MIB
 - SUN-ILOM-PET-MIB
 - SNMP-FRAMEWORK-MIB (9RFC2271.txt)
 - SNMP-MPD-MIB (RFC2572)
 - System and SNMP groups from SNMPv2-MIB (RFC1907)
 - entPhysicalTable from ENTITY-MIB (RFC2737)

Note: In this appendix, the focus is on using the CLI. For details about how to use each of the interfaces, see the appropriate sections in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Documentation Collection*.

Getting Started

- Establish a management connection to Oracle ILOM by using one of the following options:
 - Dedicated network management connection (default)
 - Sideband network management connection
 - Dedicated local management connection
 - Dedicated interconnected SP management connection
- Log in with CLI secure shell:
 1. Establish an SSH session to the Oracle ILOM CLI, open a terminal window.
 2. Log in to Oracle ILOM by using the default `root` account:

```
$ ssh root@ILOM_SP_or_CMM_ipaddress
```
 3. At the Password prompt, type `changeme`.
The Oracle ILOM CLI prompt appears (`->`).

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle ILOM firmware arrives preconfigured on your Oracle server or chassis monitoring module (CMM) in a way that makes establishing a management connection to Oracle ILOM simple and straightforward. Oracle ILOM supports the following management connections:

- **Dedicated network management connection (default):** All Oracle servers and CMMs that are shipped with Oracle ILOM provide a dedicated in-band management port on the chassis that securely segregates all management traffic away from the host. All servers and CMMs arrive ready for you to establish a secure management connection to Oracle ILOM. Simply attach an active LAN connection to the physical network management port (NET MGT) on the chassis and you are ready to log in.
- **Sideband network management connection:** For servers supporting sideband management, you can optionally connect to Oracle ILOM and manage the server remotely through the standard data port provided on the server chassis. Implementing a sideband management connection to Oracle ILOM eliminates the need to support two separate network connections for host and management traffic. However, this approach could: (1) potentially decrease the connection performance to Oracle ILOM, and (2) potentially provide risks for transmitting Oracle ILOM traffic over an untrusted network.

- **Dedicated local management connection:** All Oracle servers and CMMs arrive with a physical serial port on the chassis that makes it easy to establish a secure local management connection to Oracle ILOM. This type of management connection is particularly useful when a local console is the only way to access and diagnose system failures or when you need an alternative method for modifying the Oracle ILOM preconfigured network properties before establishing a LAN connection.
- **Dedicated interconnected SP management connection:** For Oracle servers supporting an internal Ethernet-over-USB interface, you can optionally establish a LAN management connection to Oracle ILOM from a host operating system (OS) client without the use of the network management (NET MGT) port on the server.

Note: For step-by-step instructions about how to configure each of the management connection options, see “Setting Up a Management Connection to Oracle ILOM and Logging In” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.

Oracle ILOM comes with a preconfigured user account and default network parameters that simplify logging in to Oracle ILOM for the first time. The steps for logging in through the CLI secure shell are presented in the slide. For information about how to log in through a local serial console or web browser, see “Log In To the Oracle ILOM SP or CMM” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.

Note: To exit an Oracle ILOM CLI session, type `exit`.

Agenda

- Introducing Oracle ILOM
- **Configuring host server management actions**
- Collecting system information and monitoring health status
- Setting up alert notifications and Syslog server for event logging
- Managing event and audit log entries
- Observing and debugging system behavior
- Managing Sun hardware faults through the Oracle ILOM Fault Management Shell

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuring Host Server Management Actions

With Oracle ILOM, you can configure the following SPARC host server management actions:

- Control host power to server:
CLI: `<command> /System`
- Set host diagnostic tests to run:
SP CLI: `/HOST`
- Set the boot behavior on a SPARC host server:
SP CLI: `/HOST property_name`
- Override SPARC host boot mode:
SP CLI: `/HOST/bootmode`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle ILOM enables you to perform a number of host server management actions, a subset of which are presented in the slide. Additional information about each action follows:

- **Control host power to server:** Oracle ILOM provides a set of parameters that enables you to control the power state of a host server. You can issue power control commands from the Oracle ILOM CLI or web interface. The CLI commands for SPARC include but are not limited to the following:
 - **Reset:** `reset -force /System`
 - **Graceful reset:** `reset /System`
 - **Immediate power off:** `stop -force /System`
 - **Graceful shutdown and power off:** `stop /System`
 - **Power on:** `start /System`
 - **Power cycle:** `stop /System` and `start /System`
- **Set host diagnostic tests to run:** Oracle ILOM provides a set of server-specific diagnostic properties that enable you to control whether system diagnostic tests are run at startup. These diagnostic properties are configurable from either the Oracle ILOM CLI or web interface. For a list of SPARC server SP diagnostic properties, see “Setting Host Diagnostic Tests to Run” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.

- **Set the boot behavior on a SPARC host server:** Oracle ILOM provides a set of SPARC server properties that enables you to view host control information, as well as optionally set properties to control system boot behavior. You can view host control information or set configurable SPARC server boot properties from the Oracle ILOM CLI or web interface. For a list of host control information and boot properties on a SPARC-managed server, see “Setting Boot Behavior on SPARC Host Server” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.
- **Override SPARC host boot mode:** Oracle ILOM provides a set of host boot mode properties that enables you to override the default method for booting the host operating system on the SPARC server. The host boot mode properties in Oracle ILOM are intended to help resolve corrupt boot mode settings with OpenBoot or LDoms. The boot mode properties, when set in Oracle ILOM, apply only to a single boot and expire within 10 minutes if the power on the host SPARC server is not reset. You can use the Oracle ILOM CLI or web interface to set the host boot mode properties. For a list of host boot mode properties for a host SPARC server, see “Overriding SPARC Host Boot Mode” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.

Note

- For a complete list of host server management actions you can perform with Oracle ILOM, see “Configuring Host Server Management Actions” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.
- Oracle ILOM allows you to manage SPARC diagnostics, POST, and boot mode operations by using SNMP commands. For more details and examples of commands you can run, see “Manage SPARC Diagnostics, POST, and Boot Mode Operations (SNMP)” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 SNMP, IPMI, CIM, WS-MAN Protocol Management*.

Agenda

- Introducing Oracle ILOM
- Configuring host server management actions
- **Collecting system information and monitoring health status**
- Setting up alert notifications and Syslog server for event logging
- Managing event and audit Log entries
- Observing and debugging system behavior
- Managing Sun hardware faults through the Oracle ILOM Fault Management Shell

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Collecting System Information and Monitoring Health Status

This section covers the following topics:

- Collecting information and status
- Viewing open problems

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle ILOM interfaces provide easy-to-access properties for viewing system information. For example, from the web interface Summary page or from the CLI `/System` target, you can, at a glance, collect system-specific information describing the managed device, determine the health state of the managed device, and view open problems, if detected on a managed device.

Collecting Information and Status

To collect system-level information or to verify the system health status, use the `show /System` command.

```
Properties:
  health = OK
  health_details = -
  open_problems_count = 0
  power_state = On
  locator_indicator = Off
  model = SUN FIRE X4270 M3
  type = Rack Mount
  part_number = 07011205
  serial_number = 0328MSL-1119T4002F
  system_identifier = (none)
  system_fw_version = ILOM: 3.1.0.0
  primary_operating_system = Not Available
  host_primary_mac_address = Not Available
  ilom_address = 10.123.45.255
  ilom_mac_address = 00:12:34:D5:F2:F6
  actual_power_consumption = 123 watts
  action = (none)
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The host system-level health status CLI properties are viewable at the `/System` target by using the `show /System` command, as presented in the example in the slide.

A system can have the following health state status:

- **OK:** The system or subcomponent is in good working order.
- **Service Required:** Oracle ILOM detected a problem on the managed device that will require a service action to resolve the issue. If this status appears at the system level, view the open problems detected on the managed device. If this status appears in the Open Problems table, refer to the URL provided in the table for further details.
- **Not Available:** Oracle ILOM is unable to provide a health status for this component. Oracle ILOM might require the Hardware Management Pack to be installed. For information about the Hardware Management Pack, see the note on the next page.
- **Offline:** Offline applies to the Prepare to Remove action state of a chassis subcomponent. This status appears when the action property is set to Prepare to Remove and the physical subcomponent is not physically removed from the chassis.

Note: The property value for the primary operating system installed on the managed device is shown only when the Oracle ILOM Hardware Management Pack is installed on the managed device. The Oracle Hardware Management Pack comprises two cross-platform components, Oracle Hardware Management Agents: Simple Network Management Protocol (SNMP) hardware and storage monitoring agents and Oracle Server CLI Tools, which is a family of command-line interface (CLI) tools for managing and configuring your servers. The Hardware Management Pack components enable you to:

- Use a cross-platform management agent for in-band monitoring of your Oracle hardware over SNMP, enabling you to integrate your Oracle servers into your data center management infrastructure
- Use command-line tools to configure BIOS, RAID volumes, and Oracle ILOM service processors on your servers
- Use a command-line tool to upgrade your server components
- Use IPMITool to access Oracle ILOM service processors via the IPMI protocol and perform management tasks

For more information about the Hardware Management Pack, visit the following site on the Oracle Technical Network: <http://www.oracle.com/technetwork/server-storage/servermgmt/tech/hardware-management-pack/index.html>.

You can also view subcomponent-level information and health status from the CLI by typing the following command:

```
show /System/subcomponent-category-name
```

Where *subcomponent-category-name* equals one of the subcomponent target names under *show /System*.

For example, to view server subcomponent health status for memory, type:

```
show /System/Memory
```

```
/System/Memory
```

```
Targets:
```

```
DIMMs
```

```
Properties:
```

```
health = OK
```

```
health_details = -
```

```
installed_memory = 16 GB
```

```
installed_dimms = 2
```

```
max_dimms = 16
```

```
Commands:
```

```
cd
```

```
show
```


Viewing Open Problems

If a problem occurs on a managed system, Oracle ILOM automatically:

- Illuminates the Server Action LED on the physical device
- Identifies the faulted condition in an easy-to-read Open Problems table
- Records system information about the fault condition in the event log

You can view open problems from the CLI by typing:

```
show /System/Open_Problems
```

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle ILOM automatically detects system hardware faults and environmental conditions on a managed device.

After the repair (or the replacement) of a faulty server component or a faulty Sun Blade chassis FRU, Oracle ILOM automatically clears the fault state from the Open Problems table.

Open problems are defined as follows:

- **Faulted state:** Indicates the component is present but is unusable or degraded because one or more problems have been diagnosed by Oracle ILOM. Oracle ILOM automatically disables the component to prevent damage to the system.
- **Open Problems:** Refers to the Open Problems page in the web interface or the Open Problems tabular output shown in the CLI. When a problem is detected on a managed device, Oracle ILOM identifies the problem in the Open Problems CLI output or web interface table.
- **Oracle ILOM Fault Management Shell:** Enables Oracle Services personnel to diagnose system problems and, if necessary, to override fault states. Customers should not use this shell unless requested to do so by Oracle Services.

You can view open problems detected on a host server or Blade system chassis from either the Open Problems web page or the `/System/Open_Problems` CLI target.

The Open Problems web page and the CLI target report the following information:

- Total number of problems detected
- Time stamp, name, and CLI target for each faulted component
- URL for troubleshooting a faulted component

Agenda

- Introducing Oracle ILOM
- Configuring host server management actions
- Collecting system information and monitoring health status
- **Setting up alert notifications and Syslog server for event logging**
- Managing event and audit log entries
- Observing and debugging system behavior
- Managing Sun hardware faults through the Oracle ILOM Fault Management Shell

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Setting Up Alert Notifications and Syslog Server for Event Logging

This section covers the following topics:

- Configuring alert notifications
- Configuring Syslog for event logging

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuring Alert Notifications

- Oracle ILOM supports the configuration of these types of alerts:
 - IPMI PET alerts
 - SNMP trap alerts
 - Email alert notifications
- You can configure up to 15 alert notifications with the Oracle ILOM CLI, Oracle ILOM web interface, or SNMP client.
- Oracle ILOM requires three properties to be set for each alert notification:
 - `alert type`
 - `alert destination`
 - `alert level`

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can configure alert notifications in Oracle ILOM to provide advance warnings of possible system failures. Oracle ILOM supports the configuration of IPMI PET alerts, SNMP trap alerts, and email alert notifications.

Up to 15 alert notifications are configurable in Oracle ILOM by using the Oracle ILOM CLI, Oracle ILOM web interface, or an SNMP client. For each configured alert notification, you can optionally generate a test message to ensure that the destination recipient successfully receives the test message.

Note: For instructions about how to configure, test, and disable alert notifications for IPMI PET, SNMP, and email, see “Configuring Alert Notifications” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.

For each alert notification, Oracle ILOM requires these three properties to be set: `alert type`, `alert destination`, and `alert level`. Depending on which alert type is configured, other properties are optionally configurable.

Note: For a list of alert notification properties, see “Alert Notification Configuration Properties” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 Configuration and Maintenance Guide*.

Configuring Syslog for Event Logging

- The Syslog service logs events to a remote log host.
- The events provide the following information:
 - Class
 - Type
 - Severity
 - Description
- The Syslog service is enabled by configuring a Syslog server IP address:

```
set /SP|CMM/clients/syslog \  
destination_ip=syslog_server_ip
```

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Syslog is a protocol service used for logging events to a remote log host. The events logged to a Syslog server provide all the same information that you would see in the local Oracle ILOM event log, including class, type, severity, and description.

You can enable the Syslog service in Oracle ILOM by configuring a Syslog server IP address as shown in the slide.

Oracle ILOM provides properties for configuring up to two Syslog servers.

Note: For more information about using Syslog for event logging, see “Managing ILOM Log Entries” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 User’s Guide*.

Agenda

- Introducing Oracle ILOM
- Configuring host server management actions
- Collecting system information and monitoring health status
- Setting up alert notifications and Syslog server for event logging
- **Managing event and audit log entries**
- Observing and debugging system behavior
- Managing Sun hardware faults through the Oracle ILOM Fault Management Shell

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Managing Event and Audit Log Entries

1. To list log entries from the SP or CMM CLI, perform one of the following:
 - For the event log, type either:
`show /SP/Logs/event/list`
`show /CMM/Logs/event/list`
 - For the audit log, type either:
`show /SP/Logs/audit/list`
`show /CMM/Logs/audit/list`
2. To filter the log output, use the show command and specify a value for one or more of the filter properties: Class, Type, Severity.
`show /SP|CMM/logs/event|audit/list \`
`Class==value Type==value Severity==value`
3. To clear all log entries shown, use the clear=true command:
`set /SP|CMM/logs/event|audit clear=true`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can manage event and audit log entries with the Oracle ILOM web interface or CLI. The event log tracks informational, warning, or error messages about a managed device, such as the addition or removal of a component or the failure of a component. The properties of the events recorded in the log can include the severity of the event, the event provider (class), and the date and time the event was logged.

The audit log tracks all interface-related user actions, such as user logins, logouts, configuration changes, and password changes. The user interfaces monitored for user actions include the Oracle ILOM web interface, CLI, Fault Management Shell (captive shell), and Restricted shell, as well as the SNMP and IPMI client interfaces.

Note: You need Admin (a) role privileges to clear log entries in Oracle ILOM.

The steps for managing event and audit log entries from the CLI are presented in this slide and in the next.

Note: Oracle ILOM uses UTC/GMT time zones, by default, when capturing time stamps for log entries. However, if a log file is viewed from a remote client that is located in a different time zone, Oracle ILOM automatically adjusts the time stamps in the log files to reflect the local time zone of the remote client and the host system. In this case, two time stamps appear in the log for each listed event entry.

Note for Step 1: To scroll the list, press any key except the Q key.

Note for Step 2: Where *SP* | *CMM* appears, type either *SP* or *CMM*. Where *event* | *audit* appears, type either *event* or *audit*.

Note for Step 3: When prompted, type *y* to confirm the action or *n* to cancel the action.

Agenda

- Introducing Oracle ILOM
- Configuring host server management actions
- Collecting system information and monitoring health status
- Setting up alert notifications and Syslog server for event logging
- Managing event and audit log entries
- **Observing and debugging system behavior**
- Managing Sun hardware faults through the Oracle ILOM Fault Management Shell

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle ILOM provides a collection of online and offline diagnostic tools to assist IT administrators and Oracle Services personnel who verify server behavior, troubleshoot problems, and perform repair or replacement service actions.

Observing and Debugging System Behavior

This section covers the following topics:

- Taking a snapshot of the Oracle ILOM SP state (CLI)
- Enabling SPARC diagnostics to run at boot (CLI)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Taking a Snapshot of the Oracle ILOM SP State (CLI)

1. Log in to the Oracle ILOM CLI server SP.
2. To view the snapshot properties, type:
`show SP/diag/snapshot`
3. To define the data set collection, type:
`set /SP/diag/snapshot dataset=data`
4. To define the encryption mode, type:
`set /SP/diag/snapshot
encrypt_output=true|false`
5. To start data collection, type:
`set /SP/diag/snapshot dump_uri=uri`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle ILOM Service Snapshot utility enables you to produce a snapshot of the server processor at any instant in time.

Caution: The purpose of the Oracle ILOM Service Snapshot utility is to collect data for use by Oracle Services personnel to diagnose system problems. Customers should not run this utility unless requested to do so by Oracle Services personnel.

The Oracle ILOM Service Snapshot utility gathers SP state data. The utility collects log files, runs various commands and collects their output, and sends the data collection as a downloaded file to a user-defined location.

Note: To collect SP data by using the Service Snapshot utility, you need the Admin (a) role enabled.

The steps for taking a snapshot from the CLI are presented in the slide.

Note for step 3: The data property has several value options that enable you to specify the type and amount of information you want to capture. For more information about data property values, see “Take a Snapshot of the Oracle ILOM SP State (CLI)” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 User’s Guide*.

Note for step 4: When the encryption mode is set to `true`, you must type an encryption password at the prompt in order to start data collection. Then later, you must type an encryption password at the prompt in order to decrypt the output file.

Note for step 5: The `uri` property specifies the transfer method of the output files. The URI format is as follows: `protocol://username:password@host/directory`

Where `protocol` can be one of these transfer methods: SFTP or FTP.

Note: For more information about the CLI snapshot properties, see “Take a Snapshot of the Oracle ILOM SP State (CLI)” in the *Oracle Integrated Lights Out Manager (ILOM) 3.1 User’s Guide*.

Enabling SPARC Diagnostics to Run at Boot (CLI)

1. To specify triggers for running the SPARC diagnostic tests, type:

```
set /HOST/diag trigger=value
```

2. To specify the level of diagnostics to run, perform the following:

- For when the host operating system powers on, type:

```
set /HOST/diag power_on_level=value
```

- For when the host operating system is reset by the user, type:

```
set /HOST/diag user_reset_level=value
```

- For when the host operating system is reset due to a system error, type:

```
set /HOST/diag error_reset_level=value
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On an Oracle SPARC system using Oracle ILOM, you can enable diagnostic mode, specify triggers and the level of diagnostics, as well as the verbosity of the diagnostic output.

The steps for enabling SPARC diagnostics to run at boot from the CLI are presented in this slide and the next.

Note: The Reset and Host control (r) role is required to modify the SPARC diagnostic properties in Oracle ILOM on SPARC systems.

Use the /HOST/diag host mode property to control whether diagnostics are enabled and to specify which diagnostic mode is enabled.

Note for Step 1: The *value* property can be one of the following:

- **none:** Does not run diagnostic tests
- **user-reset:** Runs diagnostics upon a user-invoked power reset
- **power-on-reset:** Runs diagnostics when power is applied to the host operating system
- **error-reset:** Runs diagnostics upon any error-invoked power reset
- **all-resets:** Runs diagnostics whenever a power reset occurs

Note for Step 2: The *value* property can be one of the following:

- **min:** Runs the minimum set of diagnostics to partially verify the health of the system
- **max (Default):** Runs the maximum set of diagnostics to fully verify the health of the system

Enabling SPARC Diagnostics to Run at Boot (CLI)

3. To specify the report verbosity when diagnostics are run, perform one of the following:
 - For when the host is powered on, type:
`set /HOST/diag power_on_verbosity=value`
 - For when the host is reset by the user, type:
`set /HOST/diag
user_reset_verbosity=value`
 - For when the host is reset due to a system error, type:
`set /HOST/diag
error_reset_verbosity=value`
4. To specify the diagnostics mode, type:
`set /HOST/diag mode=value`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note for Step 3: The *value* property can be one of the following:

- **none:** Does not print output to the system console while diagnostics are run, unless a fault is detected
- **min:** Prints limited output to the system console while diagnostics are run
- **normal (Default):** Prints a moderate amount of output to the system console while diagnostics are ran.
- **max:** Prints the full output to the system console while diagnostics are run, including the name and results for each test
- **debug:** Prints extensive debugging output to the system console while diagnostics are run, including device testing and debugging output for each test

Note for Step 4: The *value* property can be one of the following:

- **off:** Prevents the diagnostic tests from running
- **normal (Default):** Runs the diagnostic tests based upon the triggers specified in Step 1

Agenda

- Introducing Oracle ILOM
- Configuring host server management actions
- Collecting system information and monitoring health Status
- Setting up alert notifications and Syslog server for event logging
- Managing event and audit log entries
- Observing and debugging system behavior
- **Managing Sun hardware faults through the Oracle ILOM Fault Management Shell**

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Observing and Debugging System Behavior

This section covers the following topics:

- Protecting against hardware faults: Oracle ILOM Fault Manager
- Using the Oracle ILOM Fault Management Shell

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Protecting Against Hardware Faults: Oracle ILOM Fault Manager

- The Fault Manager does the following:
 - Detects and interprets errors
 - Determines whether a fault or defect is present on a managed system
- Hardware fault or defect notifications appear in the Open Problems tabular output.
- Event messages are logged to the event log and the Fault Management logs (for Oracle Services personnel).
- Oracle ILOM fault events and notifications are:
 - Cleared automatically if the repaired or replaced resource is associated with a FRU
 - Not cleared automatically if the repaired or replaced resources are not associated with a FRU

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Fault Manager in Oracle ILOM is intended to help with problems that might occur on an Oracle ILOM Sun managed device. For instance, the Fault Manager detects and interprets errors and determines whether a fault or defect is present on a managed system. When a determination is made, the Fault Manager issues a list of suspected hardware components that might be the cause of the problem.

Notifications indicating that a hardware fault or defect has been diagnosed appear in the Open Problems tabular output, which you can view from the Oracle ILOM interfaces. In addition to the hardware fault notifications provided in the Open Problems output, the Fault Manager also logs event messages to the event log and the Fault Management logs. You can view the event log from the Oracle ILOM interfaces. Oracle Services personnel can view the Fault Management logs from the Oracle ILOM Fault Management shell.

Note: When notified of a diagnosed problem, always consult the recommended knowledge article for additional details. An [http://](#) reference is provided to the recommended knowledge article in the event notification in the Open Problems output, as well in the event messages in the log files.

Fault events and notifications in Oracle ILOM are cleared automatically when the repaired or replaced resource is associated with a field-replaceable unit (FRU). When a repaired or replaced resource is not associated with a FRU, Oracle ILOM is unable to detect the repair or replacement, in which case the fault event notification is not cleared automatically in the Open Problems output or in the log files.

Using the Oracle ILOM Fault Management Shell

This section covers the following topics:

- Launching a Fault Management shell session (CLI)
- Using `fmadm faulty` to administer active Sun hardware faults
- Clearing faults for undetected replaced or repaired hardware components
- Using `fmdump` to view historical fault management logs
- Using `fmstat` to view the fault management statistics report

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle ILOM Fault Management shell enables Oracle Services personnel to view and manage fault activity detected on a Sun managed device. The purpose of the shell is to help Oracle Services personnel diagnose system problems. Although customers should not launch this shell or run fault management commands in the shell under normal circumstances, there are times when Oracle Services might ask you to do so, in which case you should be familiar with the shell and how to use it.

Launching a Fault Management Shell Session (CLI)

1. Log in to the Oracle ILOM CLI.
2. To launch a Fault Management shell session, type:
`start /SP/faultmgmt/shell`
3. Run the appropriate Fault Management shell commands to perform any of the following:
 - Administer active faulty components (`fmadm faulty`).
 - View historical fault management activity (`fmdump`).
 - View a statistical report of fault management operations (`fmstat`).
4. To display help information for one of the following external commands, type `help fmadm`, `help fmdump`, or `help fmstat`.
5. To exit the Fault Management shell, at the `faultmgmt` prompt, type `exit`.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The steps for launching a session from the CLI are presented in the slide and should be performed only when directed to do so by Oracle Services.

Note: Admin (a) role privileges are required to launch the Fault Management shell from the Oracle ILOM CLI.

Note for step 2: One of the following Fault Management shell command prompts appears:

- `faultmgmtsp>` appears for Sun SP managed devices.
- `faultmgmtcmm>` appears for Sun CMM managed devices.

After you start the Fault Management Shell, and until you exit the Fault Management Shell, you can issue only commands that are specific to the Fault Management Shell.

Note for step 3: For instructions about how to perform each of these tasks, see the slides that follow.

Note for step 5: To issue standard Oracle ILOM CLI commands, you must first exit the Fault Management Shell.

Using `fmadm` to Administer Active Sun Hardware Faults

1. To view information about active faulty hardware components reported for a Sun managed device, type:
`fmadm faulty <-display_option>`.
2. When applicable, refer to the `http://` referenced knowledge article in the `fmadm faulty` output for further instructions for resolving a reported problem.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To view and manage active Sun hardware faults that are conventionally maintained by the Oracle ILOM Fault Manager, you use the `fmadm` utility in the Fault Management Shell. The steps for viewing information about active faulty components from the Fault Management Shell CLI are presented in the slide.

Note: For Sun hardware customers, the preferred method for viewing active information about faulty components is to view the health state of a component in the Open Problems tabular output, which is provided in the Oracle ILOM CLI and web interface.

Note for step 1: The display options for the `fmadm faulty` command are as follows:

- `fmadm faulty -a`: Displays all active faulty components
- `fmadm faulty -f`: Displays active faulty FRUs
- `fmadm faulty -r`: Displays active fault FRUs and their fault management states
- `fmadm faulty -s`: Displays one-line fault summary for each fault event
- `fmadm faulty -u <uuid>`: Displays fault diagnosis events that match a specific universal unique identifier (UUID)

Clearing Faults for Undetected Replaced or Repaired Hardware Components

1. Identify and display information about active suspect components by using the `fmadm faulty` command.
2. To manually clear a fault for an undetected replaced or repaired hardware components, type the appropriate repair commands.
3. To display the exit code for the last executed fault management command, type:
`echo $?`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With permission from Oracle Services personnel, you can issue `fmadm` repair commands from the Oracle ILOM Fault Management Shell to manually clear fault messages for undetected repair or replacement service actions. The steps for clearing faults for undetected hardware repairs or replacement are presented in the slide.

Note: If a fault event is cleared before completing the corrective service action required for the faulty component, the Oracle ILOM Fault Manager diagnoses the fault and redisplay the fault event in the Oracle ILOM Open Problems table, as well as in the Oracle ILOM Fault Management log files.

Note for step 1: See the previous slide about “Using `fmadm` to Administer Active Sun Hardware Faults” for instructions on how to view component information.

Note for step 2: The repair commands are as follows:

- To indicate that a suspect component or uuid resource has been replaced or removed, type:
`fmadm replaced <fru|cru|uuid>`

- To indicate that a suspect component or uuid resource has been physically repaired to resolve the reported problem (for example, reseating a component or fixing a bent pin), type:
`fmadm repaired <fru|cru|uuid>`
- To indicate that a suspect component or uuid resource is not the cause of the problem, type:
`fmadm acquit <fru|cru|uuid>`
Where `<fru|cru|uuid>` appears, type the system path to the suspect chassis FRU or CRU; or type the associated universal unique identifier (uuid) for the resource reported in the problem.

A replacement takes precedence over a repair, and both a replacement and a repair take precedence over an acquittal. Therefore, you can acquit a component and then subsequently repair it, but you cannot acquit a component that has already been repaired.

Note for step 3: One of the following echo codes appears:

- 0: Indicates successful completion
- 1: Indicates an error has occurred. Errors include a failure to communicate with Oracle ILOM or insufficient privileges to perform the requested operation.

Using `fmdump` to View Historical Fault Management Logs

1. Display the contents maintained in a Fault Management log file set by using the `fmdump` command.
2. Rotate the log display by using either `fmadm rotate fltlog` or `fmadm rotate errlog`.
3. Display the exit code for the last executed fault management command:

```
echo $?
```

Caution: Do not base administrative service actions on content in the Fault Management historical log files, but rather on the active `fmadm faulty` output. The Fault Management log files contain historical events, which should not be considered active events for faults or defects.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle ILOM Fault Manager maintains historical information about system problems in two sets of log files for Oracle Services personnel's use. A log file set can consist of active system events together with a possible number of older system events. The steps for viewing the Fault Management log files are presented in the slide and should be performed only with permission from Oracle Services.

Note for step 1: Your options are as follows:

- To display the fault log, type:
`fmdump`
The Fault Management fault log records human-readable fault diagnosis information and the problems possibly related to the symptoms. A time stamp and description are provided for each event recorded.
- To display a fault log for a specific universal unique identifier (uuid), type:
`fmdump -u <uuid>`
- To display the error log, type:
`fmdump -e`

The Fault Management error log records error telemetry and the symptoms of problems detected by the system. Each problem recorded identifies:

- A time stamp for when the problem was detected
- A universal unique identifier (UUID) that uniquely identifies a particular problem across any set of systems
- An `http://` identifier that provides access to a corresponding knowledge article posted on the Oracle support website

For the fault log, in particular, it is important to recognize that `fmddump` shows all problems ever diagnosed and is not limited to active problems diagnosed. To view active faults only, issue the `fmadm faulty` command.

Note for step 3: One of the following echo codes appears:

- 0: Indicates successful completion. All records in the log file were examined successfully.
- 1: Indicates that invalid command-line options were specified

Using `fmstat` to View the Fault Management Statistics Report

To view the Fault Management statistics report, run the `fmstat` command.

```
faultmgmtsp> fmstat
fdd statistics      2013-03-03/19:12:51

engine              status      evts_in  evts_out  errors
repair              empty        8         0         0
hysteresis          empty        0         0         0
SERD                 empty        0         0         0
simple               empty       12         0         0
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle ILOM Fault Manager maintains a viewable statistics report about diagnosis engines and agents participating in fault management operations. You use the `fmstat` command to view statistics for fault management operations.

Note: This task should be performed only with permission from Oracle Services.

An example of a report is presented in the slide. The report property descriptions are as follows:

- **engine:** Identifies the name of the diagnosis engine:
 - **Repair:** Rule that indicates a fault should be considered repaired if a specified ereport is logged. For example, the fault `fault.chassis.power.inadequate@/sys` would be considered repaired if `ereport.chassis.boot.power-off-requested@/system` was logged.
 - **Hysteresis:** Rule to diagnose a fault if ereport *A* (initiation) is logged and ereport *B* (cancellation) is not logged within some specified time afterward. For example, ereport *A* is `ereport.fan.speed-low-asserted` and ereport *B* is `ereport.fan.speed-low-deasserted`. The time limit between the initiation/cancellation can be no greater than 10 seconds.