Automatically Installing Oracle® Solaris 11.4 Systems

Part No: E60976
August 2021

Automatically Installing Oracle Solaris 11.4 Systems

Part No: E60976

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle recognizes the influence of ethnic and cultural values and is working to remove language from our products and documentation that might be considered insensitive. While doing so, we are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is an ongoing, long-term process.

Référence: E60976

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info ou le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs si vous êtes malentendant.

Contents

Using This Documentation

- **Overview** – Describes how to install the current Oracle Solaris operating system using the automatic installer (AI).
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Advanced experience troubleshooting and replacing hardware

Product Documentation Library

Documentation and resources for this product and related products are available at https://docs.oracle.com/cd/E37838_01/.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Overview of Installation Options

The Oracle Solaris software can be installed in a number of different ways depending on your needs. The chapter covers the following topics:

What's New in Installation for Oracle Solaris 11.4

- First boot services and packages can now be created using the `svc-create-first-boot` command. Administrators can quickly create first boot SMF services that can be set up by Automated Installer (AI). See "Automatically Creating a First-boot Service and Package" in *Customizing Automated Installations With Manifests and Profiles*.
- For SPARC clients, the WAN Boot protocol can use the HMAC-SHA 256 algorithm to enhance network installation security. The `installadm` command becomes the administrative tool for HMAC key generations for both SPARC and x86 clients.

  On UEFI-enabled x86 systems, support for secure boot establishes a chain of trust from early boot through the entire installation process. The administrator can associate keys and certificates in the BIOS that can be used for securing the initial boot mechanism all the way to contacting the AI and IPS package repository.

  For more information about securing network installation, see Chapter 5, "Securing Automated Installation".
- With the migration of persistent network configuration to SMF, you can apply complex network settings to the installation process. This feature especially benefits automated installation operations. In a system configuration profile, you can assign values to SMF network properties to be implemented during AI. See Chapter 3, "Working With System Configuration Profiles" in *Customizing Automated Installations With Manifests and Profiles*.

Installation Methods and the Installation Documentation

To install Oracle Solaris, two general methods are available:

- Using the text installer
- Using Automated Installer (AI)

In turn, each method has options for further customizing how your chosen installation method would run in your specific environment.

Based on these available methods, the documentation for Oracle Solaris installation is organized as follows:

- To use the text installer, refer to *Manually Installing an Oracle Solaris 11.4 System*. It describes procedures for installing Oracle Solaris manually.
- To use the automated installer (AI), refer to *Automatically Installing Oracle Solaris 11.4 Systems*, which is the current guide. It describes procedures to set up the necessary components for a "hands-free" Oracle Solaris installation.
- To perform a customized automated installation, refer to *Customizing Automated Installations With Manifests and Profiles*. This guide discusses in further detail how to use AI manifests and system configuration files to customize an automated installation. It is an important companion guide especially to *Automatically Installing Oracle Solaris 11.4 Systems*.

Additional Installation Options

The following additional installation options are supported:

Creating custom installation images	Installations are based on default installation images. However, you can build a custom image based on any of the default images. The distribution constructor tool enables you to specify parameters for building a new image. See *Creating a Custom Oracle Solaris 11.4 Image*.
Cloning an Oracle Solaris system	Through the Unified Archives feature, you can clone an existing Oracle Solaris system and use that image as a basis for the installation. See *Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4*.
Updating an installed Oracle Solaris system	You cannot use the installer tool to update existing Oracle Solaris systems. Instead, you use the `pkg` utility to access package repositories

and download new or updated software packages for your system. For further information, see *Updating Your Operating System to Oracle Solaris 11.4* and *Updating Systems and Adding Software in Oracle Solaris 11.4*.

System Requirements for OS Installations

To check the minimum memory, disk space, and other system requirements for installing the Oracle Solaris 11.4 release, see *Oracle Solaris 11.4 Release Notes*.

Make sure that your system's firmware is updated to the latest version before installing Oracle Solaris 11.4. See `https://www.oracle.com/servers/technologies/firmware.html`.

On SPARC systems, the firmware must be updated to include the current version of the Open Boot PROM (OBP) that contains the latest WAN boot support.

To test whether a SPARC system can be a WAN boot client, issue the following command:

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

If the variable `network-boot-arguments` is displayed, or if the command returns the output `network-boot-arguments: data not available`, the SPARC system's OBP supports WAN boot. Otherwise, you must use alternatives such as using media. See Chapter 8, “Automated Installations That Boot From Media”.

Note - Non Oracle x86 systems with Intel® Virtualization Technology for Directed I/O (VT-d) must have the `Intel VT-d` parameter set to `Enabled` before you install Oracle Solaris on those systems. Refer to their respective documentation for instructions to set this parameter.

About IP Addresses in This Guide

IP addresses that are used in Oracle Solaris 11 documentation conform to RFC 5737, IPv4 Address Blocks Reserved for Documentation (`https://tools.ietf.org/html/rfc5737`) and RFC 3849, IPv6 Address Prefix Reserved for Documentation (`https://tools.ietf.org/html/rfc3849`). IPv4 addresses used in this documentation are blocks 192.0.2.0/24, 198.51.100.0/24, and 203.0.113.0/24. IPv6 addresses have prefix 2001:DB8::/32.

To show a subnet, the block is divided into multiple subnets by borrowing enough bits from the host to create the required subnet. For example, host address 192.0.2.0 might have subnets 192.0.2.32/27 and 192.0.2.64/27.

♦ ♦ ♦ CHAPTER 2

Automated Installation of Multiple Systems

Automated installation is best suited for environments with multiple and mixed systems of SPARC and x86 platforms. The chapter covers the following topics:

- "About Automated Installation"
- "Components of the Automated Installer"
- "Using Rights Profiles to Install Oracle Solaris"

About Automated Installation

Automated installation uses Oracle Solaris's Automated Installer (AI) to execute a "hands-free" OS installation and client-specific configuration. Two methods of automated installation are available:

- Using an install media. This method does not require an AI server. You use the media containing the AI image to boot the system and start the installation. See Chapter 8, "Automated Installations That Boot From Media".
- Using an AI server to install on multiple systems. This method requires more preparations to set up the AI server.

Components of the Automated Installer

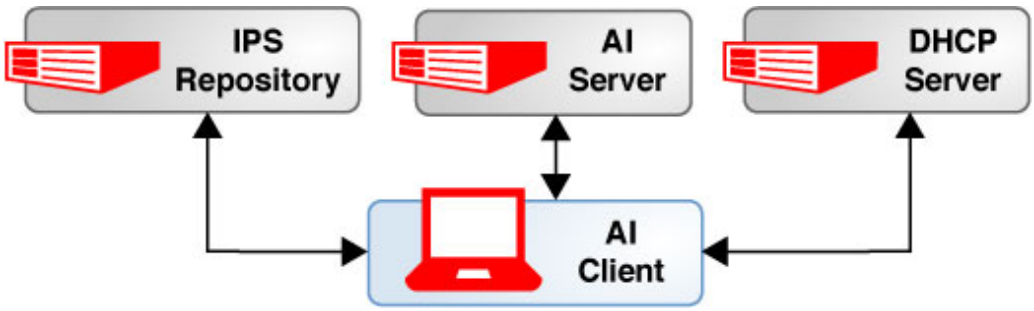
Automated installation consists of the following components:

- An AI server that manages all the processes and files that are involved in automated install operations.
- A DHCP server that provides network connectivity between the client and the AI components.

- One or more Image Packaging System (IPS) repositories of software packages to be installed.

The components are illustrated in the following figure.

FIGURE 1 AI Network Example



An initial decision you would make in using AI is the location of these components. All three can be hosted on a single system. Co-locating particularly the AI server and the DHCP server in a common system greatly eases the administration of AI operations.

AI Server

The AI server is the software in the `installadm` package that executes the AI program. The package is included when you install Oracle Solaris. The AI server is composed of the following components:

- One or more install services
- One or more AI manifests.
- Optional system configuration profiles.
- Optionally, a first-boot script to perform additional configurations that cannot be managed by the manifests or the configuration profiles.

Install Services

The install service installs the operating system based on defined parameters. The AI server can manage multiple install services to serve multiple clients. An install service is mapped to a

platform-OS version combination, such as Oracle Solaris 11.2 on SPARC, Oracle Solaris 11.3 on x86, Oracle Solaris 11.4 on SPARC, and so on.

AI Manifests

An AI manifest contains the parameters that define how install services install the OS, such as client provisioning, disk layout, packages to add, and so on. An install service must have at least one associated AI manifest.

You can create as many manifests as needed, each with different customized instructions to direct how the installation is performed on clients.

Clients can share multiple manifests. However, a client can use only one manifest at a time in one instance of installation.

Note - By default, AI with default settings does not include the Gnome desktop. To include the desktop in an AI, manually add the appropriate package to the list in the installation manifest, under the `<software_data>` section. For example:

```
<sofware_data action="install">
   other-packages
   <name>pkg:/group/system/solaris-desktop</name>
</software_data>
```

For more information about creating specialized AI manifests, see Chapter 2, "Working With AI Manifests" in *Customizing Automated Installations With Manifests and Profiles*. The chapter includes examples of manifests with customized contents.

System Configuration Profiles

System configuration profiles enable you to automate post-installation configurations, which you would otherwise have to perform manually, such as setting time zones, name services, host names, and so on.

Just as with AI manifests, you can create multiple system configuration profiles to be applied to different target clients.

For more information about creating and defining parameters in configuration profiles, see Chapter 3, "Working With System Configuration Profiles" in *Customizing Automated Installations With Manifests and Profiles*.

DHCP Server

The DHCP server manages network connections between the AI components and the clients. The DHCP server informs the AI server of the location of the CGI file to support WAN Boot clients.

To set up the DHCP server on the same system as the AI server, see "Setting the AI Server to Manage DHCP" on page 29.

Unlike an x86 client, a SPARC client can be configured to locate the AI server without DHCP. For alternative ways to perform AI without DHCP, see "Starting Automatic Installation on Clients" on page 70.

IPS Repositories

The IPS repository contains the software packages to be installed. You can also add to the repository first-boot scripts that complete client configuration at the completion of AI.

As a best practice, create repositories locally at your site. Local repositories provide advantages such as security as well as faster connectivity, which in turn makes AI operations efficient. The repository can be on the same system as the AI server or on another system on the local network.

This document does not discuss publishers and repositories. These are discussed in detail in the following guides:

- *Creating Package Repositories in Oracle Solaris 11.4*
- *Updating Systems and Adding Software in Oracle Solaris 11.4*

Unified archives are alternatives to packages in IPS repositories. Make sure that these unified archives are network accessible. To work with unified archives, see *Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4*.

Using Rights Profiles to Install Oracle Solaris

Oracle Solaris implements role-based access control (RBAC) to control system access. To perform specific tasks and run privileged commands on the system, you must have the profiles that provide you the authorization.

The following list shows some of the profiles that need to be assigned to you to install Oracle Solaris.

- Install Client Management enables you to install Oracle Solaris on client systems.
- Install Manifest Management enables you to create or configure manifests to customize the installation.
- Install Profile Management enables you to create and configure system configuration profiles to customize the installation.

Some profiles are supersets of a combination of profiles. For example, the Install Service Management profile contains the three profiles in the previous list.

The list of required profiles expands if you perform additional tasks that might be indirectly connected to your current one, such as network configuration or zone configuration.

An administrator that has the `solaris.delegate.*` authorization can assign the necessary profiles to users to enable them to perform administrative tasks in Oracle Solaris.

For example, an administrator assigns the Install Service Management rights profile to user `jdoe`. Before `jdoe` executes a privileged installation command, `jdoe` must be in a profile shell. The shell can be created by issuing the `pfbash` command. Or, `jdoe` can combine `pfexec` with every privileged command that is issued, such as `pfexec installadm`.

As an alternative, instead of assigning profiles directly to users, a system administrator can create a role that would contain a combination of required profiles to perform a range of tasks.

Suppose that a role `installadmin` is created with the profiles for installation as well as for zone creation and configuration. User `jdoe` can issue the `su` command to assume that role. All roles automatically get `pfbash` as the default shell.

For more information about rights profiles, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

♦ ♦ ♦ CHAPTER 3

Getting Started With AI Operations

This chapter describes how to plan AI operations at your site. It covers the following topics:

Planning the AI Implementation

This section assumes that you have already determined the location of the AI server as described in "Components of the Automated Installer" on page 15. To properly use AI, other factors need to be considered. The following questions serve as guidelines:

- Are your target clients using the same platform?

  If your data center uses a hybrid of system platforms, then the clients are automatically divided into two groups based on two supported platforms: SPARC and x86. Each group must have its own install service.
- Do you intend all your clients to use the same Oracle Solaris version?

  If you are installing different Oracle Solaris versions, then your platform based groups are further subdivided according to these versions. Each of those groups also requires its own install service.
- Within specific groups, are there differences among clients that warrant further subgroupings, such as disk layout or partitions, package requirements, and so on?

  If clients within groups have different installation requirements, then you would use multiple manifests that contain instructions specific to their respective target client groups.
- Within the groups, do your target clients have certain system identity characteristics that the other clients do not share?

  For example, some target clients might need different time zone configuration, or use default language other than the default English, and so on. In this case, you would need to create system configuration profiles to automatically configure these clients accordingly. The configurations take place when the system reboots after the installation is completed.

- Are there other post installation configurations that profiles cannot execute automatically?

  You can use first-boot scripts to run these configurations.

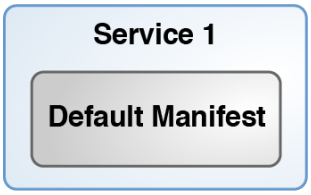
By properly identifying groups of clients, you can determine how many install services, manifests, and configuration profiles you would need. See the examples of AI setups in the next section.

For more details on customizing manifests and profiles, see *Customizing Automated Installations With Manifests and Profiles*.

Examples of Automated Install Setups

The following figures illustrate different AI server configurations. The configuration you adopt depends on your environment.

FIGURE 2 One AI Service Supporting One Architecture and One OS



This configuration is the simplest to create and assumes an environment where clients share the same architecture and OS version. Clients are booted from the network which starts the installation. After each system is rebooted, an interactive tool prompts you for system configuration information.

FIGURE 3 Two Install Services Supporting Two Architectures

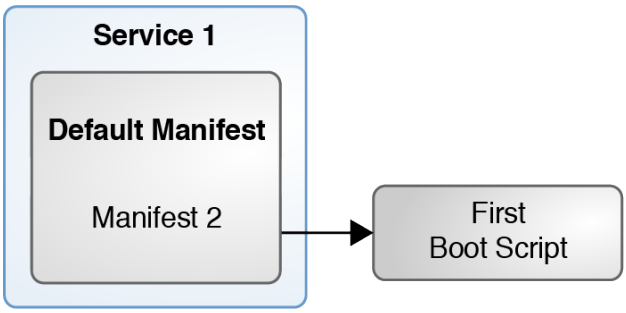
AI requires that each architecture must have its own install service. As you create the install services, their corresponding default manifests are also created. System configuration is performed manually.

This setup also applies in a situation where you have two OS versions to install on two groups of clients, even though they share the same architecture.

FIGURE 4 AI Service Supporting Two Types of Installation With Different Installation Parameters

In this scenario, some clients require an installation that is different from the rest in the group. Thus, a separate manifest with special instructions exists to cater to these clients. The rest would use the default manifest. System configuration is performed manually.

FIGURE 5 AI Service Supporting Specific Clients That Use a Script

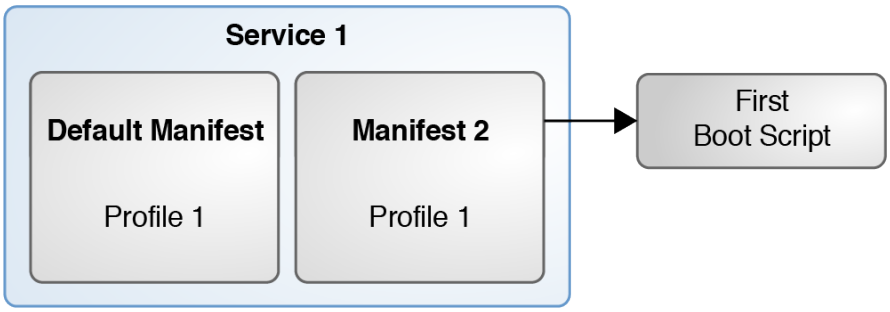


This example is a slight variation of Figure 4 in the previous example. The clients that use `Manifest 2` also use a script that configures the system after it reboots at the end of the installation. For the other clients that use the default manifest, system configuration is performed manually.

FIGURE 6 One AI Service Supporting Clients With Different Post-Install Configurations

In this scenario, the OS installation is the same for all clients. However, one group requires a different system configuration from the rest, such as the time zone, types of network configuration, and so on. Thus, separate system configuration profiles are created for each group that automatically runs after the clients are rebooted at the end of the installation.

FIGURE 7 AI Server Supporting Many Configuration Settings



This example shows how you can set up AI depending on the complexities of the needs of the clients. You can combine services, manifests, profiles, and boot scripts that will address the specific requirements of different clients in the data center. In scenarios such as this, client criteria must be well defined so that the correct manifest and profile definitions or boot scripts are applied to the appropriate clients.

Overview of Automated Installation Tasks

The configuration of automated installation involves the following general tasks.

- On the AI server
  - Configure the AI server to also manage DHCP to support AI.
  - Create install services to service SPARC and x86 clients.
  - If necessary, create manifests and configuration profiles to serve different client groups.
  - Create and associate clients with services as needed.
  - Create first-boot scripts. This task is optional. See Chapter 6, “Running a Custom Script During First Boot” in *Customizing Automated Installations With Manifests and Profiles* for more information.
- Prepare the client systems as needed.

These steps are further explained in Chapter 4, “Setting Up the AI Server” and Chapter 6, “Running AI on Clients by Using an AI Server”.

♦ ♦ ♦ CHAPTER 4

Setting Up the AI Server

To service clients over the network, AI requires a separate system to function as an AI server. This chapter describes the configuration of the AI server and its components. It covers the following topics:

- "Server Configuration Procedures (Task Map)"
- "Configuring an AI Server"
- "Working With Install Services"
- "Administering the AI Server and its Components"

Server Configuration Procedures (Task Map)

To use AI requires configurations most especially on the server. This table lists the main tasks to perform.

Task	Description	For More Information
Prepare the AI server system.	Configures the system that functions as the AI server to manage automated installs.	"Configuring an AI Server" "Setting the AI Server to Manage DHCP" Optional AI configurations: ■ "Configuring the AI Server's Web Server Files Directory" ■ "Generating AI Telemetry Information" ■ Chapter 5, "Securing Automated Installation"
Configure AI services.	Creates the services that execute the installation processes on client systems.	"Creating and Configuring an Install Service"
Configure AI manifests.	Creates manifests that direct the installations on clients.	"About AI Manifests":

Task	Description	For More Information
		Also, from *Customizing Automated Installations With Manifests and Profiles*: ■ "Working With AI Manifests" ■ "Associating AI Manifests With Install Services"
Assign configuration profiles to services.	Creates profiles that apply post-installation configurations to clients.	"About Configuration Profiles" Also, from *Customizing Automated Installations With Manifests and Profiles*: ■ "Working With System Configuration Profiles" ■ "Specifying Criteria for AI Manifests and System Configuration Profiles" ■ "Associating System Configuration Profiles With Install Services" ■ For KMIP client installations: "Extracting Configuration Information for KMIP Clients"
Associate clients to install services.	Assigns clients to services based on specific configurations of those clients.	"Creating Client-Service Associations"

Configuring an AI Server

Any x86 or SPARC system, including laptops, desktops, and virtual machines can function as an AI server:

Requirements for an AI Server

The system where the AI server is configured must meet the following hardware requirements:

- At least 1 GB of memory.
- Sufficient disk space for all the install services you intend to create. Each net image in an install service is approximately 300-400 MB.

The software setup must also be configured as follows:

- The system is running Oracle Solaris 11.4.

To upgrade a system that is running a previous release, see *Updating Your Operating System to Oracle Solaris 11.4*. To perform a fresh install, see *Manually Installing an Oracle Solaris 11.4 System*.

- The system has a static IP address.

  If no static IP address exists, issue commands as shown in the following example.

  ```
  $ ipadm create-ip net0
  $ ipadm create-addr -a 192.0.2.0/24 net0
  ```

  Type `ipadm` to display network interface information.
- The AI package is installed.

  Type `pkg list installadm` to verify.

  Note - The AI package is included in a regular Oracle Solaris installation. If the package is missing, add it with the command `pkg install install/installadm`.

- The appropriate install images, if needed, are downloaded.

  The download site is at `https://www.oracle.com/solaris/solaris11/downloads/solaris-downloads.html`.

Additional AI Server Configurations

The following are optional configurations of the AI server.

For examples of `installadm` command options, see "Selected Options for Setting Up the AI Server" on page 37. For more details, see the `installadm(8)` man page.

Setting the AI Server to Manage DHCP

The `install set-server` command with the following options sets the AI server to manage DHCP:

- `-m` enables the AI server to manage DHCP.
- `-i` sets the start of IP addresses to manage.
- `-c` sets the count or range of addresses to manage.

This command syntax sets the DHCP's managed address range from 203.0.113.10 to 203.0.113.29:

```
$ installadm set-server -m -i 203.0.113.10 -c 20
```

Configuring the AI Server's Web Server Files Directory

Note - This section assumes that the AI server and the web server are running on the same system. This guide does not cover web server configuration. To set up a web server, consult other documentation, such as `http://httpd.apache.org/docs/`.

The web server's directories serve net images, AI manifests, system configuration profiles, and other files needed for installation.

The following properties of the `svc:/system/install/server:default` SMF service identify the locations for the web server's user files:

- `all_services/webserver_files_dir` for files that do not need to be secure.
- `all_services/webserver_secure_files_dir` for files that need to be secure.

To define the location of public files, use this set of commands:

```
$ svccfg -s svc:/system/install/server:default
svc:/system/install/server:default> setprop \
   all_services/webserver_files_dir = astring: public-location
svc:/system/install/server:default> refresh
svc:/system/install/server:default> quit
```

To define the location of secure files, use this set of commands:

```
$ svccfg -s svc:/system/install/server:default
svc:/system/install/server:default> setprop \
   all_services/webserver_secure_files_dir = astring: secure-location
svc:/system/install/server:default> refresh
svc:/system/install/server:default> quit
```

Tip - For greatest security, files in the `webserver_secure_files_dir` directory should be owned by user `webservd` and group `webservd` and have no world access.

To view the directories, use one of the following URLs:

- `http://`*AI-server*`:5555/files` for unsecured files.
- `https://`*AI-server*`:5556/secure_files` for secured files.

The *AI-server* can be a host name or an IP address. You can use port numbers other than the default ones. See Configuring the AI Web Server Port Number (-p Option) and Example 11, “Configuring the Secure AI Web Server Port Number (-P Option),” on page 38.

If the AI manifest specifies an IPS package repository that requires a certificate and key, you can store those publisher credentials, and then specify this URI in the AI manifest. Only clients that have security credentials assigned can access this directory.

Enabling Multicast DNS

Multiple AI servers can exist on the same subnet. To make these servers accessible to clients, enable multicast DNS (mDNS).

See also "Multicast DNS and Service Discovery" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

▼ How to Enable Multicast DNS on the AI Server

Before You Begin Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. **(Optional) Verify that the mDNS package is installed.**

```
$ pkg info dns/mdns
```

If the package is absent, use the `pkg install` command to add it to the system.

2. **Update the name service switch information.**

Add `mdns` to the configuration of the `config/host` property.

```
$ /usr/sbin/svccfg -s svc:/system/name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: "files dns mdns"
svc:/system/name-service/switch> select system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch> quit
```

3. **Enable the mDNS service.**

```
$ svcadm enable svc:/network/dns/multicast:default
```

Generating AI Telemetry Information

To enable the server to generate telemetry information about AI, use these commands:

- `installadm set-server --telemetry-enable` switches on the telemetry feature.

- `installadm set-server --telemetry-frequency` *integer* specifies the frequency in seconds that telemetry data from the client is returned to the AI server for you to monitor. A zero (0) value means data is returned immediately. By default, this frequency is set to 120 seconds.

For the two commands, you can use these options:

- `--telemetry-success` specifies telemetry-related files the client feeds the AI server if the installation completes successfully.
- `--telemetry-failure` specifies telemetry-related files the client feeds the AI server if the installation fails.

  For both options, specify one of the following values:

  - `none` – no files are returned.
  - `intall_log` – the `/system/volatile/install_log` is returned.
  - `all_logs` – in addition to the `/system/volatile/install_log`, other various SMF install logs that were created during installation are returned.
  - `all_files` – all the files that are listed in `all_logs` as well as the additional AI install files that are used to install the system are returned.
  - *files* – files specified by the system administrator are returned.
  - *smf-fmri* – the log file for the specified SMF service FMRI is returned.

For example, this command sets the server to generate all logs if failure occurs:

```
$ installadm set-server --telemetry-enable --telemetry-frequency 90 \
   --telemetry-failure all_logs
```

Set the AI server for telemetry information before you start the installation. Transmission of telemetry information begins when the client is booted with the `boot_archive` file.

Setting Secure Shell for Remote Monitoring

Enable remote access to the server so you can connect to it and observe an ongoing installation.

Enabling Remote Access to x86 Clients

For x86 clients, enable remote access on the clients' install service. For example:

```
$ installadm set-service -n svcname-for-x86 -b livessh=enable
```

The `livessh` property is stored in the `/etc/netboot/`*svcname-for-x86*`/grub.cfg` file:

```
$multiboot $kern /platform/i86pc/kernel/amd64/unix -B livessh=enable,...
```

Enabling Remote Access to SPARC Clients

For SPARC clients, enable remote access on the service's `/etc/netboot/`*svcname-for-sparc*`/system.conf` file. See lines in bold in the following example:

```
$ cat /etc/netboot/svcname-for-sparc/system.conf
...install_service=svcname-for-sparc
install_svc_address=$serverIP:5555
livessh=enable
...
```

To enable `ssh` on SPARC clients individually, specify `livessh` on the `boot` command line when you boot the system. See “Starting Automatic Installation on Clients” on page 70.

Working With Install Services

An install service runs the automated installs of Oracle Solaris.

Creating and Configuring an Install Service

To create an install service, use the following command:

```
installadm create-service options [-y]
```

The `-y` option suppresses command prompts. Besides the `-y` option, the following are some other typical options:

- `-s` *full-path* specifies the location of the IOS image to be used for the service.
- `-d` *full-path* specifies the destination directory where the install service's files are stored.
- `-n` *service-name* enables you to specify a preferred name for the service instead of using the default name.

Note - You must create a separate service for each platform-OS version combination that describes your target clients.

If you create the service without using options, the command does the following:

- Names the service with the convention `solaris`*OS-version-architecture*, based on the AI system's OS version and architecture, for example, `solaris11_4-i386`.
- Installs the image in `/export/auto_install/`*service-name* directory.
- If this is the first service created, also creates a `default-`*architecture* service.

The `installadm list` command displays the results:

```
$ installadm create-service -y
$ installadm list
Service Name     Status Arch Type Secure Alias Aliases Clients Profiles Manifests
------------     ------ ---- ---- ------ ----- ------- ------- -------- ---------
default-i386     on     i386 pkg  no     yes   0       0       0        1
solaris11_4-i386 on     i386 pkg  no     no    1       0       0        1
```

These examples show additional options for creating the install service.

EXAMPLE 1 Creating a SPARC Install Service Using an ISO File

The `-s` option specifies a source for the image to be used for the service. The example assumes that the AI server is a SPARC system.

```
$ installadm create-service -s /var/tmp/images/sparc/sol-11_4-ai-sparc.iso
  0% : Service svc:/network/dns/multicast:default is not online. Installation services
       will not be advertised via multicast DNS.
...
100% : Created Service: 'solaris11_4-sparc'
...
```

EXAMPLE 2 Creating an Install Service for a Different Architecture

This example uses the `-a` option to specify the architecture for the install service. Use this option if the service's clients and the AI server do not share the same architecture. You must also use the `-n` option to assign a name to the service. This option applies only if you are creating the service from an IPS package.

```
$ installadm create-service -n solaris11_4-i386 -a sparc -y
```

EXAMPLE 3 x86: Creating a Service That Automatically Installs an X86 Client

By default, AI does not automatically start on x86 clients. The `-b` option enables you to override the default behavior.

```
$ installadm create-service -s /var/tmp/images/i386/sol-11_4-ai-x86.iso \
```

```
-y -b install=true
```

EXAMPLE 4 Setting the Image Path for an Install Service

The install image is normally located in `/export/auto_install/`*service-name*. The `-d` option sets the location of the image path elsewhere.

```
$ installadm create-service -d /export/ai-images -n solaris11_4-i386
```

About AI Manifests

When you create an install service for the first time, a default manifest file is also automatically created. All services use this manifest as a default file for installing Oracle Solaris on all valid clients. Valid clients are those whose architecture matches that for which the service was created. The setup is displayed as follows:

```
$ installadm list
Service Name Status Arch Type Secure Alias Aliases Clients Profiles Manifests
------------ ------ ---- ---- ------ ----- ------- ------- -------- ---------
ai-x86       on     i386 pkg  no     no    1       0       0        1
default-i386 on     i386 pkg  no     yes   0       0       0        1
```

With this minimum setup, you can already start an automated install on clients by simply booting the clients from the network.

However, in more complicated scenarios with different installation requirements, you would need multiple manifests with their own instructions. For more information about creating various manifests with customized configuration instructions, see Chapter 2, “Working With AI Manifests” in *Customizing Automated Installations With Manifests and Profiles*.

About Configuration Profiles

Configuration profiles enable you to automate post-installation configurations which you would otherwise perform manually.

A typical tool for creating configuration profiles is the System Configuration Interactive (SCI) tool. For example, to create a profile in `/var/tmp/`, you would type:

```
$ sysconfig create-profile -o /var/tmp/
```

The command opens the SCI tool and displays screens where you set parameter values such as time zones, language, and so on. The configurations are saved in `sc_profile.xml` in the directory you specified.

The profile must be assigned to an install service in order to be used in an installation. For example:

```
$ installadm create-profile -p myprofile -f /var/tmp/sc_profile.xml -n ai-x86
```

The `-p` option creates the profile instance for the service `ai-x86`. When you run the installation, system configuration based on the profile becomes part of the automated operation.

If an install service does not have any associated configuration profile, then you would have to perform the post-installation configuration manually. When the client reboots after the completion of the installation, the SCI tool automatically opens. You specify values on different screens to configure the system.

For additional information about system configuration profiles, see Chapter 3, "Working With System Configuration Profiles" in *Customizing Automated Installations With Manifests and Profiles*.

Creating Client-Service Associations

Default install services are automatically associated with clients that match the platform for which the install services are created.

If clients need to use an install service other than the default service, then you must manually associate the server and those clients. The clients are identified by their MAC addresses, as shown in the following example:

```
$ installadm create-client -e aa:bb:cc:dd:ee -n ai-x86
```

A client can be associated with only one install service at one AI instance. If you run the command on the same client multiple times for different services, that client is associated only with the install service that was specified last.

To remove an association, delete the client:

```
$ installadm delete-client -e mac-address
```

Every creation of a client also modifies the DHCP configuration file of the DHCP server. Thus, the DHCP server recognizes and provides network connectivity to the client.

After you have properly created and configured the services, manifests, profiles, and server-client associations as needed, you can proceed with the installation on the clients. See Chapter 6, "Running AI on Clients by Using an AI Server".

The examples that follow show additional ways to associate clients to install services.

EXAMPLE 5 Setting Up a SPARC Client With an Install Service

The following example associates the SPARC client with MAC address `00:14:4f:a7:65:70` with a service called `sparc-service01`.

```
# installadm create-client -n sparc-service01 -e 00:14:4f:a7:65:70
```

EXAMPLE 6 x86: Associating an X86 Client With an Install Service and Redirecting Output to a Serial Line

In this example, the installation output is redirected to a serial console device.

```
$ installadm create-client -e c0ffeec0ffee -n solaris11_4-i386 -b console=ttya
```

EXAMPLE 7 x86: Changing Boot Properties for an X86 Client

The `-G` option is specific to x86 clients to specify a custom GRUB2 menu to use when booting the client. In this example, the custom GRUB2 menu is `/etc/netboot/grub.custom.cfg`.

```
$ installadm create-client -e c0ffeec0ffee -n solaris11_4-i386 -G /etc/netboot/
grub.custom.cfg
```

Note that the `-b` and `-G` options cannot be used at the same time.

Administering the AI Server and its Components

This section describes more options for administering and maintaining the AI server and the install services that the AI server hosts.

Selected Options for Setting Up the AI Server

These examples show how to use different options to set up the AI server. The examples assume that your role has the appropriate rights profile to issue the commands. See “Using Rights Profiles to Install Oracle Solaris” on page 18.

EXAMPLE 8 Disabling AI Support on a Network (`-L` Option)

By default, the AI server is configured to serve clients on all networks to which the system is connected. In this example, the `192.0.2.0/24` network is removed from servicing AI requests.

```
$ installadm set-server -L 192.0.2.0/24
```

EXAMPLE 9 Including Networks to Be Supported by an AI Server (`-l` Option)

The following command shows how to allow install services on two networks.

```
$ installadm set-server -l 192.0.2.0/24, 198.51.100.0/24
```

EXAMPLE 10 Configuring the AI Web Server Port Number (`-p` Option)

An AI server hosts install services by using a web server. By default, the AI web server is hosted on port 5555. The following command changes the port number:

```
$ installadm set-server -p 7000
```

EXAMPLE 11 Configuring the Secure AI Web Server Port Number (`-P` Option)

A secure AI server hosts install services by using a web server. By default, the secure AI web server is hosted on port 5556. The following command changes the port number:

```
$ installadm set-server -P 7001
```

EXAMPLE 12 Configuring the Default Image Path (`-d` Option)

By default, images for all install services in the AI server are created in `/export/auto_install/`*service-name*. The following command changes the default location.

```
$ installadm set-server -d /export/aiimages
```

To override the default directory location for a specific service, use the `-d` option when creating the service. For example:

```
$ installadm create-service -d /var/tmp -n myservice
```

EXAMPLE 13 Disabling Automatic Updates of the Local DHCP Service on an AI Server (`-M` Option)

By default, the local ISC DHCP configuration is automatically updated when client and service configurations are modified in the AI server. This example shows how to disable automatic maintenance of the local ISC DHCP configuration.

```
$ installadm set-server -M
Changed Server
Disabling SMF service svc:/network/dhcp/server:ipv4
Refreshing SMF service svc:/system/install/server:default
```

Updating an Existing Install Service

The `update-service` subcommand updates the image associated with a service alias. The subcommand creates a new service with the updated image. The existing alias is reassigned to the new service.

The `-p` option specifies a different repository for the image with which the service is updated. If the option is not specified, the same repository as when the service was first created is used. If you specify a publisher, then as an option, you can also specify the key and cert that are needed to create or update that service. See Example 15, “Using a Different Repository When Updating an Install Service,” on page 39.

If the `-s` option is not specified, the newest available version of the `install-image/solaris-auto-install` package is used from the publisher.

EXAMPLE 14 Updating an Install Service

Suppose that the following services exist in the AI server.

```
$ installadm list
Service Name      Base Service       Status Arch  Type Ali Cli Man Pro
------------      ------------       ------ ----  ---- --- --- --- ---
default-i386      solaris11-i386     on     i386  pkg  0   1   1   0
solaris11-i386    -                  on     i386  pkg  1   0   1   0
```

You update the service:

```
$ installadm update-service -n default-i386

$ installadm list
Service Name      Base Service         Status Arch  Type Ali Cli Man Pro
------------      ------------         ------ ----  ---- --- --- --- ---
default-i386      solaris11-1-i386     on     i386  pkg  0   1   1   0
solaris11-i386    -                    on     i386  pkg  1   0   1   0
solaris11-1-i386  -                    on     i386  pkg  1   0   1   0
```

The new service with the updated image is `solaris11-i386-1` and becomes the base service of the `default-i386` alias.

EXAMPLE 15 Using a Different Repository When Updating an Install Service

This example shows how to identify the publisher associated with the `solaris11_4-i386` service. First type `installadm list -v` to determine the image path for the service. Then use the image path to identify the current publisher.

Note that the option to specify the key or cert can be used with either the `installadm update-service` command or the `installadm create-service`.

```
$ installadm list -v -n solaris11_4-i386
Service Name      Status Arch  Type Alias Aliases Clients Profiles Manifests
------------      ------ ----  ---- ----- ------- ------- -------- ---------
solaris11_4-i386  on     i386  iso  no    1       0       1        1

   Image Path ........ /export/auto_install/solaris11_4-i386
    ....

$ pkg -R /export/auto_install/solaris11_4-i386 publisher
PUBLISHER        TYPE     STATUS   URI
solaris          origin   online   http://pkg.oracle.com/solaris/release/
```

This example updates the default service's publisher to `example.com/solaris/mybuild`.

```
$ installadm update-service -n default-i386 \
   -p solaris=http://example.com/solaris/mybuild \
...[--key path-to-pem-formatted-key \
    --cert path to pem-formatted-certificate]
```

EXAMPLE 16 Using a Different Net Image Package When Updating an Install Service

This example specifies a specific net image package.

```
$ installadm update-service -n default-i386 -s FMRI
```

Showing Information About Install Services

The `installadm list` command shows information about install services and associated clients, AI manifests and system configuration profiles.

Note - For security related information, see "Listing Security Information Related to AI" on page 56.

EXAMPLE 17 Listing All Install Services on the AI Server

This example displays all of the install services on this AI server. In this example, four enabled install services are found. Disabled services have an `off` Status value. To obtain information only about a specific service, add the `-n` *service* option in the command.

```
$ installadm list
```

```
Service Name      Base Service       Status Arch  Type Ali Cli Man Pro
------------      ------------       ------ ----  ---- --- --- --- ---
default-i386      solaris11-i386     on     i386  pkg  0   1   1   0
default-sparc     solaris11_4-sparc  on     sparc pkg  0   0   0   1
solaris11-i386    -                  on     i386  pkg  1   0   1   0
solaris11_4-sparc -                  on     sparc pkg  1   0   1   2
```

EXAMPLE 18 Listing Clients Associated With Install Services

This example lists all the clients that are associated with the install services on this AI server. The clients were associated with the install services by using the `installadm create-client` command. See "Creating Client-Service Associations" on page 36.

To obtain client information for only a specific service, add the `-n` *service* option in the command.

```
$ installadm list -c

Service Name      Client Address    Arch  Secure Custom Args Custom Grub
------------      --------------    ----  ------ ----------- -----------
solaris11_4-sparc 00:14:4F:A7:65:70 sparc no     no          no
solaris11_4-i386  08:00:27:8B:BD:71 i386  no     no          no
                  01:C2:52:E6:4B:E0 i386  no     no          no
```

EXAMPLE 19 Listing All AI Manifests and System Configuration Profiles

This example lists all AI manifests, derived manifest scripts, and system configuration profiles for all install services on this AI server. The Service and Manifest Name and Profile Name columns display the internal names of the manifests, scripts, or profiles. The Status column identifies the default manifest for each service and any inactive manifests. A manifest is inactive if it does not have any associated criteria and also is not the default. The Criteria column shows the associated client criteria.

Manifests that are named `orig_default` indicate that they are created by a script that creates default manifests for default services.

To obtain information for only a specific service, add the `-n` *service* option in the command.

```
$ installadm list -m -p
Service Name                   Manifest Name      Type      Status    Criteria
------------                   -------------      ----      ------    --------
default-i386                   orig_default       derived   default   none
default-sparc                  orig_default       derived   default   none
solaris1132-i386               ipv4               xml       active    ipv4 =
                                                                      198.51.100.0/24
                                                                      - 192.0.2.0/24
```

```
                                mem1              derived   default   (Ignored: mem =
                                                                      2048 MB -
                                                                      4095 MB)
                                orig_default      derived   inactive  none
solaris11_4-sparc               sparc-ent         xml       active    mem = 4096 MB
                                                                      - unbounded
                                                                      platform =
                                                                      SUNWSPARC-
                                                                      Enterprise
                                mem1              derived   default   (Ignored: mem =
                                                                      2048 MB -
                                                                      4095 MB)
                                orig_default      derived   inactive  none

Service Name                    Profile Name  Environment Criteria
------------                    ------------  ----------- --------
solaris11_4-i386                mac2          system      mac = 08:00:27:8B:BD:71
                                                          hostname = server2
                                mac3          system      mac = 01:C2:52:E6:4B:E0
                                                          hostname = server3
                                ipv4          system      ipv4 = 203.0.113.0/24 -
                                                          198.51.100.0/24
                                mem1          system      mem = 2048 MB - 4095 MB
solaris11_4-sparc               mac1          system      mac = 01:C2:52:E6:4B:E0
                                                          hostname = server1
                                                          ipv4 = 192.0.2.0/24
                                sparc-ent     system      platform =
                                                          SUNWSPARC-Enterprise
                                                          mem = 4096-unbounded
```

If you run this command with the rights profile, an additional column in the list of manifests identifies the type of the manifest, either `xml` or `derived`.

♦ ♦ ♦ CHAPTER 5

Securing Automated Installation

This chapter describes how to configure security for automated installation over the network. It covers the following topics:

Overview of Securing Automated Installations

You can secure automated installations with the Transport Layer Security (TLS) protocol. TLS uses private certificates and key pairs as well as the Certificate Authority (CA) certificate for generating and signing certificates. SPARC WAN boot clients also require firmware hash (HMAC) digest and encryption keys which secure the downloading of the initial network boot files.

The current Oracle Solaris release supports HMAC-SHA256 protocols for SPARC WAN boot clients, in addition to the HMAC-SHA1 protocols in previous releases.

Note - With x86 clients that use PXEBoot, the initial network boot phase is not secured. For these clients, you implement security by creating an install service from a custom image that has security credentials. You would set the same credentials for the service as the image's. For information about creating custom secured AI images, see Chapter 3, "Building an Image" in *Creating a Custom Oracle Solaris 11.4 Image*.

Security for automated installations is implemented in the following ways:

- Server and client authentication.
- Access control access to automated installations and server data.
- Client data protection either for all clients together or separately for specific clients.
- Data encryption.
- Secure access to IPS package repositories as well as user specified directories.

In addition, you can also use AI to provision Kerberos in the clients. See “How to Configure Kerberos Clients Using AI” on page 59.

Commands for Securing Automated Installation

To secure automated installations, you use `installadm` subcommands and security related options. The subcommand corresponds to the component or entity you are securing. The subcommand to use also depends on whether you are setting security while creating the entity or you are configuring an existing entity.

- `set-server` for the AI server
- `create-service` or `set-service` for install services
- `create-client` or `set-client` for both SPARC and x86 clients

Generating Credentials

You can generate credentials either automatically or by providing user-supplied credentials.

- To generate credentials automatically, use the `-g` option. For example:

  ```
  $ installadm set-server -g
  ```

  In this example, the `-g` option generates or regenerates HTTPS credentials as well as firmware keys. See “Securing Automated Installations” on page 45 for other examples of how to generate credentials.
- To provide the credentials yourself, use the following options:
  - `-A` specifies the path to the PEM-encoded X.509 Certificate Authority (CA) certificate file.

    CA certificates must have unique subject lines. You specify each CA chain of trust one time. If the CA chain includes more than one CA certificate file, use separate `-A` options.

- `-C` specifies the path to a PEM-encoded X.509 certificate file.
- `-K` specifies the path to a PEM-encoded X.509 private key file. This key file must have any passphrase removed.

In the following example, all 3 certificate files are created at one time.

```
$ installadm set-server -C server.crt -K server.key -A cacert.pem
```

The certificate authority (CA) can be created separately from the certificate and key files. However, the certificate authority must be created first before you create the certificate and key files. However, the certificate and key files must be created together.

Configuring Security for Automated Installations (Task Map)

The following table lists the general tasks for securing automated installation.

Task	Description	For More Information
Configure AI server security.	Creates security credentials for the AI server.	"Securing AI on the AI Server"
Configure install services security.	Generates security credentials for install services.	"Configuring Install Service Credentials"
Create service security policy.	An optional task that sets the policy for determining how clients access the AI server.	"Setting a Service Policy"
Configure client security.	An optional task that creates security credentials for specific client systems.	"Configuring Client Credentials"
Modify AI manifest.	An optional task to add definitions to an AI manifest to use a secure IPS repository for the specific client.	"Modifying the AI Manifest to Install From a Secure IPS Repository"
Set security keys on clients.	For SPARC clients only: set security keys for SPARC clients.	"Configuring WAN Boot Security for SPARC Clients"

Securing Automated Installations

To assign security credentials, use the following command format.

```
$ installadm set-entity [-D] -f|--hmac-type signature-type \
   [-g| [-H|--generate-hmac-key]]
```

set-*entity*

Specifies the subcommand to use depending on the component or entity you are configuring: `set-server`, `set-client`, or `set-service`.

`-D`

Changes the default client security credentials. This option is used only with the `set-server` subcommand.

`-f` or `--hmac-type` *signature-type*

Sets the signature type for the server, client, or service. It can be either `hmac-sha1` or `hmac-sha256`. If you specify `hmac-sha1`, the signature type applies only to SPARC clients. For x86 clients, `hmac-sha256` is the only supported type.

`-g`

Generates or regenerates HTTPS credentials. The option also generates firmware keys if these do not exist. The HMAC key that is generated is based on the signature type you specified.

`-H` or `--generate-hmac-key`

Regenerates existing HMAC firmware keys according to the signature type you specified. Note that the `-H` option is for key regeneration only. An error occurs if you use the option while no keys actually exist.

Note - If you are servicing SPARC clients, then after you generate HMAC keys, you must also set those keys on the individual client's firmware. See “Configuring WAN Boot Security for SPARC Clients” on page 52.

The sections that follow show how to apply this command to the AI server, install services, and specific clients.

Securing AI on the AI Server

On the AI server, you can create credentials as well as set the HMAC policy for WAN boot clients.

Setting the HMAC Policy

Configuring the HMAC policy sets the server-wide policy for HMAC signatures. After you have set the policy, the HMAC type becomes the default signature for new custom clients or

services. You do not need to specify the policy again unless you set an exception to the policy, or if you are performing upgrades.

Setting the policy does not affect existing SPARC clients and services until you specifically set their firmware keys. For information about upgrading security on SPARC clients, see "Upgrading Security Credentials" on page 54.

This example sets the policy to use HMAC-SHA256 as the default signature for future clients and services:

```
$ installadm set-server --hmac-policy hmac-sha256
Changed Server.
```

Note - You can also use the shorter form of the option:

```
$ installadm set-server -F hmac-sha256
```

Setting Credentials for the AI Server

Use the following command:

```
$ installadm set-server [-D] --hmac-type signature-type [-g| [-H]]
```

The `-D` option is only used with the `set-server` subcommand. For an explanation of the options, see "Securing Automated Installations" on page 45.

If you use only the `--hmac-type` *signature-type* option, the command sets the active HMAC signature type for the server.

The following example automatically creates HTTP certificates. Because the `--hmac-type` is not specified, then for SPARC clients, HMAC keys are generated based on the default HMAC-SHA1 protocol.

```
$ installadm set-server -g
The root CA certificate has been generated.
The CA signing certificate request has been generated.
The signing CA certificate has been generated.
A new certificate key has been generated.
A new certificate has been generated.
Generated client encryption (AES) firmware key:
   8d210964e95f2a333c5e749790633273
Generating new hashing key (HMAC)...
Generated client hashing (HMAC SHA-1) firmware key:
```

```
   4088861239fa3f3bed22f8eb885bfa476952fab4
Changed Server
```

To generate credentials as well as SHA256 keys, you would type the following command instead:

```
$ installadm set-server --hmac-type hmac-sha256 -g
```

Note - To manually provide your own credentials, see "Generating Credentials" on page 44.

Configuring Install Service Credentials

For an AI service, in addition to generating security credentials, you can also optionally set a service policy to determine how the service is accessed.

Setting Credentials for an AI Service

To generate security credentials for an install service, use the following command:

```
$ installadm set-service -n service-name --hmac-type signature-type [-g| [-H]]
```

For an explanation of the different options, see "Securing Automated Installations" on page 45.

Note - To provide your own credentials instead, see "Generating Credentials" on page 44.

Setting a Service Policy

The `-p` option sets an authentication policy for a service. Each install service can have one set security policy. Select from the following choices:

`require-client-auth`

Confirms the identity of the client. Requires client and server authentication for all clients of the specified service. This option also requires encryption.

Requires all clients of the service to authenticate with client authentication. All clients of the specified service must be assigned credentials, and all SPARC clients of this service

must have their firmware keys generated. Any clients of the service that are not configured for client authentication will not be able to use this install service.

`require-server-auth`

Confirms the identify of the AI server. Requires all clients of the specified service to perform server authentication. This option also requires encryption.

Requires at least AI server authentication for access to the specified install service. Client authentication is optional, but you must provide any assigned or attributed client credentials. You must also define firmware keys for all clients of this service.

`optional`

Allows both authenticated and unauthenticated clients to access the install service. The option also requires encryption if the AI server has credentials. This is the default behavior.

You must provide any assigned client credentials. Clients without assigned or attributed credentials do not use firmware keys or server authentication. Server authentication is provided only for clients configured for client authentication.

`encr-only`

For x86 clients only: Enables SSL/TLS end-to-end encryption without requiring authentication. Without authentication, the identities of the client and AI server are not guaranteed. Data in transit is not readable over the network by third parties.

`disable`

Disables all security for all clients of the specified service.

Clients of this service are not authenticated. No credentials are issued. Clients of this service cannot access the `webserver_secure_files_dir` directory described in "Configuring the AI Server's Web Server Files Directory" on page 30. Use this setting with caution: Any install service files that were previously protected by authentication are no longer protected. Client data is not secured from unwanted access. To re-enable authentication, specify the `set-service` subcommand again with a different security policy value.

The following additional examples show how you can set security for the install service.

EXAMPLE 20 Requiring AI Server Authentication During Installation

This example specifies a security setting that requires server authentication to use an install service. Use the `require-server-auth` install service security setting to require clients of the specified service to at least authenticate the AI server.

```
$ installadm set-service -p require-server-auth -n install-service
```

EXAMPLE 21 x86: Requiring Encryption During Installation

This example specifies a security setting that uses encryption but does not require authentication. On x86 clients, to protect data transfers for a specific install service but not require client or server authentication, use the `encr-only` security setting. You still need a server certificate. The data will be protected from snooping over the network, but the AI server will provide the data to any client that issues the proper request to the server.

```
$ installadm set-service -p encr-only -n install-service
```

Configuring Client Credentials

Client security provides the following benefits:

- The AI server can verify the identity of the clients.
- Data is encrypted over the network.
- For clients with custom credentials, any published files specific to a client are not readable by any other client.
- Only authenticated clients can access the user-specified secure directory described in "Configuring the AI Server's Web Server Files Directory" on page 30.

To configure security for a specific client, use the following command:

```
$ installadm set-client -e mac-address --hmac-type signature-type [-g| [-H]]
```

For an explanation of the other options, see "Securing Automated Installations" on page 45.

Note - When you move a client from one install service to another, the client's custom credentials are unaffected. To associate clients with a service, see "Creating Client-Service Associations" on page 36.

EXAMPLE 22 Assigning User-Supplied Credentials for Specific Clients

This example specifies user-supplied credentials. Firmware keys are generated if they do not already exist and are displayed on screen.

```
$ installadm set-client -e 02:00:00:00:00:00 -C client.crt -K client.key -A cacert.pem
```

For an explanation of the options, see "Securing AI on the AI Server" on page 46.

EXAMPLE 23 Setting Credentials for Clients of a Specific Install Service

Non-custom clients use the credentials of their associated AI service. See the following example for the `solaris11_4-sparc` service.

```
$ installadm set-service -g -n solaris11_4-sparc
Generating credentials for service solaris11_4-sparc...
A new certificate key has been generated.
A new certificate has been generated.
Generated client encryption (AES) firmware key:
   34bc980ccc8dfee478f89b5acbdf51b4
Generated client hashing (HMAC SHA-1) firmware key:
   b8a9f0b3472e8c3b29443daf7c9d448faad14fee
```

Clients without credentials that are assigned to the service share the service's credentials. Consequently, these clients can view each other's installation data.

EXAMPLE 24 Setting Default Client Credentials

To provide a default set of credentials for any client, you configure the credentials on the AI server and use the `-D` option.

```
$ installadm set-server -D -g
Generating default client credentials...
A new certificate key has been generated.
A new certificate has been generated.
Generated client encryption (AES) firmware key:
   7cdbda5b8fc4b10ffbd29fa19d13af77
Generated client hashing (HMAC SHA-1) firmware key:
   14effe2c515da4940ef1db165791e92790163004
```

After default client credentials are assigned, all clients would perform client and server authentication, and firmware keys are required for all the clients.

Because multiple clients share the same default credentials, they can view each other's installation data.

Regenerating Firmware Keys

To regenerate encryption and hashing keys, use the `-E` and `-H` options respectively. The existing encryption key or HMAC key would be invalidated and replaced. You can regenerate these keys separately or together by specifying both options in the single command.

The HMAC type affects the behavior of the `-H` key regeneration option. If you regenerate the keys without specifying the HMAC type, existing hash keys are destroyed, and only the hash key for the current HMAC type configuration for that service or client is generated.

Consider the following command use cases:

- `$ installadm set-`*entity* `-H`

  If the current HMAC type is SHA1, the command destroys current keys and generates SHA1 keys.

- `$ installadm set-`*entity* `-f hmac-sha256 -H`

  If the current HMAC type is SHA1, the command destroys current keys and generates SHA256 keys.

Note that the `-H` option is for key regeneration only. An error would occur if you use the option while no keys actually exist.

Note - If you have SPARC WAN boot clients, make sure that every time you regenerate firmware keys, you also update the encryption and hash keys on those clients accordingly. Otherwise, those clients cannot use secure installation.

SPARC: Configuring WAN Boot Security for SPARC Clients

To install Oracle Solaris on SPARC clients over the network, you must set firmware keys on those clients. You set the keys by issuing commands on the OBP command prompt.

- `set-security-key wanboot-aes` *key* sets the encryption key.
- `set-security-key wanboot-hmac-sha1` *key* sets the SHA1 firmware key.
- `set-security-key wanboot-hmac-sha256` *key* sets the SHA256 firmware key.

Caution - Using these same commands without specifying values for *key* deletes corresponding keys in the client. This deletion would prevent a SPARC client from using install services that have been configured to require server and client authentication.

▼ How to Secure WAN Boot on SPARC Clients

Before You Begin Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. List the security information for clients.

```
$ installadm list -v
```

If you created separate credentials for a specific SPARC client, include the client's MAC address in the command to obtain its client's security information. For example:

```
$ installadm list -ve aabbccddeeff
```

2. Note down the AES key and active hash key for the client.

In the following example, the active hash key is based on the HMAC-SHA256 algorithm.

```
$ installadm list -vs
...
Def Client FW Encr Key ........
   31c88df08c958972a4b0996910539a39
Def Client FW HMAC-SHA1 Key ... (inactive)
   3789ec373712f89879c575643415b386564b0e51
Def Client FW HMAC-SHA256 Key . (active)
   ae956c3a41d02083ca40f6125fce994d5df4a3e5077f9996d6118dce5ac74fad
HMAC Policy ................... HMAC-SHA256
```

3. On the client system, access the OpenBoot prompt.

Several options exist to access the OpenBoot prompt, such as typing the command `init 0`.

If the `auto-boot?` OpenBoot variable is set to `false`, rebooting the system also displays the `ok` prompt at the end of the boot process.

4. At the OBP prompt, set the AES key and the active hash key.

```
ok set-security-key wanboot-aes 31c88df08c958972a4b0996910539a39
ok set-security-key wanboot-hmac-sha256 \
   ae956c3a41d02083ca40f6125fce994d5df4a3e5077f9996d6118dce5ac74fad
```

If the active security key is HMAC-SHA1, then you would use the appropriate command argument:

```
ok set-security-key wanboot-hmac-sha1 key
```

SPARC: Upgrading Security Credentials

Upgrading SPARC clients using SHA1 protocols to use the preferred SHA256 protocol can be completed in phases. When you set the policy on the AI server to use SHA256, keys that would be generated for future clients would be based on that protocol. Likewise, when you set the HMAC type of an install service to SHA256, this type determines the keys that you subsequently generate for future clients to use. You would then set these keys on those SPARC clients from the OBP command prompt.

However, existing SPARC clients that use SHA1 keys continue to use these keys until you reset them to use the new protocol.

When you a generate a new SHA256 hash key, the `installadm` functionality stores and maintains the SHA1 and SHA256 keys internally. Depending on the `hmac-type`, one key is rendered active while the other is deactivated. You can display the information about both active and inactive keys with the `installadm list` command.

▼ SPARC: How to Upgrade a Client's HMAC Key Based on its Service

This task shows you how to upgrade SPARC clients that use SHA1 keys to switch to its service's SHA256 keys. It assumes that no SHA256 keys have yet been set up on the install service.

Before You Begin Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. **If necessary, check the clients that use the SPARC service.**

```
$ installadm list -c
Service Name      Client Address    Arch  Secure Custom Args Custom Grub
------------      --------------    ----  ------ ----------- -----------
solaris11_4-sparc A0:B1:C2:D3:E4:F5 sparc yes     no          no
```

   In this example, the system with the MAC address A0:B1:C2:D3:E4:F5 uses the `solaris11_4-sparc` service.

2. **(Optional) Check the current HMAC key that the client uses.**
   For example:

```
$ installadm list -v -e A0:B1:C2:D3:E4:F5
Service Name         Client Address    Arch  Secure Custom Args Custom Grub
```

```
------------         --------------    ----  ------ ----------- -----------
solaris11_4-sparc    A0:B1:C2:D3:E4:F5 sparc yes    no          no

...

   FW Encr Key (AES) . 23780bc444636f124ba3ff61bdac32d1
   FW HMAC Key (SHA1) 1093562559ec45a5bb5235b27c1d0545ff259d63
   Boot Args ......... none
```

3. **On the AI server, create SHA256 keys.**
   Perform one or both substeps depending on the security configuration you want to implement.

   a. **Create SHA256 keys on the service.**

   ```
   $ installadm set-service -n solaris11_4-sparc --hmac-type sha256
   Assigning credentials for service solaris11_4-sparc...
   Generating new hashing key (HMAC)...
   Generated service hashing (HMAC SHA-256) firmware key
      b8a9f0b3472e8c3b29443daf7c9d448faad14feeb795895dac7a36d4ba6e1084
   ```

   b. **Create SHA256 keys on a client.**

   ```
   $ installadm set-client -g -e aa:bb:cc:aa:bb:cc -hmac-type sha256
   Assigning credentials for client AA:BB:CC:AA:BB:CC...

   Generating new hashing key (HMAC)...
   Generated client hashing (HMAC SHA-256) firmware key:
      b795895dac7a36d4ba6e1084e906aa24fda9c973e7fb4ee1c55199ca50825d3f
   Changed Client A0:B1:C2:D3:E4:F5
   ```

   Both steps perform the following actions:

   - Create new SHA256 keys.
   - Set the new keys as the active keys.

4. **Access the client system and set the new key on the its firmware.**
   Based on the previous step, you would do one or both of the following steps:

   a. **Based on Step 3a, you would type the following on the client A0:B1:C2:D3:E4:F5:**

   ```
   OK set-security-key wanboot-hmac-256 \
      b8a9f0b3472e8c3b29443daf7c9d448faad14feeb795895dac7a36d4ba6e1084
   ```

   b. **Based on Step 3b, you would type the following on the client AA:BB:CC:AA:BB:CC:**

```
OK set-security-key wanboot-hmac-256 \
   b795895dac7a36d4ba6e1084e906aa24fda9c973e7fb4ee1c55199ca50825d3f
```

Listing Security Information Related to AI

To display security-related information, you use the `installadm list` command combined with available options. The following examples show the types of information you can obtain.

For displaying other AI information that is not related to security, see "Showing Information About Install Services" on page 40.

Displaying AI Server Security Information

To display server security information, use the `-v` and `-s` options. In addition to what is typically displayed in previous Oracle Solaris versions, the output includes additional HMAC information (displayed in bold in the example):

```
$ installadm list -sv
AI Server Parameter        Value
-------------------        -----
Hostname ................. hostname.example.com
Architecture ............. sparc
Active Networks .......... 192.0.2.0/24
                           198.51.100.0/24
Http Port ................ 5555
Secure Port .............. 5556
Image Path Base Dir ...... /export/auto_install
Multi-Homed? ............. yes
Managing DHCP? ........... yes
DHCP IP Range ............ 198.51.100.0/24 - 203.0.113.0/24
Boot Server .............. 192.0.2.0/24
Web UI Enabled? .......... yes
Wizard Saves to Server? .. no
Security Enabled? ........ no
Security Key? ............ yes
Security Certificate:
              Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=hostname.example.com
              Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
              Source : Server Certificate
              Valid from:  Dec 12 19:48:00 2016 GMT
                      to:  Dec 10 19:48:00 2026 GMT
```

```
              Validates?: yes
CA Certificates:
     f9d73b41 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
              Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
              Source : Server CA Certificate
              Valid from:  Dec  8 10:31:00 2016 GMT
                      to:  Dec  6 10:31:00 2026 GMT
     d09051e4 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
              Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
              Source : Server CA Certificate
              Valid from:  Dec  8 10:31:00 2016 GMT
                      to:  Dec  6 10:31:00 2026 GMT
Def Client Sec Key? ........... yes
Def Client Sec Cert:
              Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Client default
              Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
              Source : Default Client Certificate
              Valid from:  Feb  2 16:47:00 2017 GMT
                      to:  Jan 31 16:47:00 2027 GMT
              Validates?: yes
Def Client CA Certs ........... none
Def Client FW Encr Key ........
   31c88df08c958972a4b0996910539a39
Def Client FW HMAC-SHA1 Key ... (inactive)
   3789ec373712f89879c575643415b386564b0e51
Def Client FW HMAC-SHA256 Key . (active)
   ae956c3a41d02083ca40f6125fce994d5df4a3e5077f9996d6118dce5ac74fad
HMAC Policy ................... HMAC-SHA256
Number of Services ............ 25
Number of Clients ............. 3
Number of Manifests ........... 26
Number of Profiles ............ 3
Telemetry Enabled? ............ no
Telemetry Success:
     none
Telemetry Failure:
     all_logs
Telemetry Frequency ........... 120 second(s)
Telemetry Files Retention ..... 1 year(s)
Telemetry Statistics Retention  1 year(s)
```

Displaying Client Security Information

To display client configuration information particularly with reference to security, use the -v and -e options. The output is similar to the server security information except for the HMAC policy, which is set only on the AI server.

```
$ installadm list -ve abcdefabcdef
Service Name Client Address    Arch  Secure Custom Args Custom Grub
------------ --------------    ----  ------ ----------- -----------
case_02      AB:CD:EF:AB:CD:EF sparc no     no          no

...

  Security Key? ..... yes
  Security Cert:
       Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=CID 01ABCDEFABCDEF
       Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
       Source : Client Certificate
       Valid from:  Mar 24 10:24:00 2017 GMT
                to:  Mar 22 10:24:00 2027 GMT
       Validates?: yes
  CA Certificates:
    d09051e4 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
       Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
       Source : Default CA Certificate
       Valid from:  Dec  8 10:31:00 2016 GMT
                to:  Dec  6 10:31:00 2026 GMT
  FW Encr Key (AES) .
   f444b7415cfbeadc3121e6dc42c77d3d
  FW HMAC-SHA1 Key .. (inactive)
   368954d00efa469b223bc88aaa62ea994292727e
  FW HMAC-SHA256 Key  (active)
   b795895dac7a36d4ba6e1084e906aa24fda9c973e7fb4ee1c55199ca50825d3f
  Boot Args ......... none
```

Displaying Other Client Security Information

The `installadm export` command lists information about TLS and X.509 keys that have been configured on a client system.

The `-C` option displays the client's x.509 TLS certificate.

```
$ installadm export -e ab:cd:ef:ab:cd:ef -C
------ certificate: client_AB:CD:EF:AB:CD:EF_cert_de22916b ------
-----BEGIN CERTIFICATE-----
MIICFDCCAX+gAwIBAgIBGTALBgkqhkiG9w0BAQswUDELMAkGA1UEBhMCVVMxDzAN
 ....
UiZDA6GOdvE=
-----END CERTIFICATE-----
```

The `-K` option displays the client's X.509 private key:

```
$ installadm export -e ab:cd:ef:ab:cd:ef -K
--------------- key: client_AB:CD:EF:AB:CD:EF_key ---------------
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDCCJbC5Bd0uMQ0AOk4lLlQqWiQwqkx9lpIhHl31tF1/WxHi74A
 ...
SYoBeKAOPSo7Evund+bHAROl0H4QnbSJgl1UDuZr3T3h
-----END RSA PRIVATE KEY-----
```

Provisioning Kerberos Clients

In addition to securing the AI process, you can also use AI to provision Kerberos in clients.

▼ How to Configure Kerberos Clients Using AI

In this procedure, the keytab file for the Kerberos client has already been created and stored on the AI server. In the examples use auto-registration to configure Kerberos clients by using preexisting credentials or using new principals. The auto-registration process is simpler because you do not have to create and encode keytab files for individual Kerberos clients.

Before You Begin Ensure that your role has the appropriate rights profiles to perform this procedure. See “Using Rights Profiles to Install Oracle Solaris” on page 18.

1. **Create an install service, if needed.**

```
$ installadm create-service -n krb-sparc \
   -d /export/auto_install/krb-sparc \
   -s /export/auto_install/iso/sol-11_4-ai-sparc.iso
Creating service from:
/export/auto_install/iso/sol-11_4-ai-sparc.iso
Setting up the image ...
Creating sparc service: krb-sparc
Image path: /export/auto_install/krb-sparc
Refreshing install services
```

2. **Associate Kerberos clients with a service.**

   Repeat this step for all clients that need to be installed running Kerberos. In this example the client using the address of `11:11:11:11:11:11` is associated with the `krb-sparc` install service.

```
$ installadm create-client -n krb-sparc -e 11:11:11:11:11:11
Adding host entry for 11:11:11:11:11:11 to local DHCP configuration.
```

3. **Create credentials for the Kerberos clients.**

```
$ installadm set-client -c 11:11:11:11:11:11 -g
Generating credentials for client 11:11:11:11:11:11...
A new certificate key has been generated.
A new certificate has been generated.
```

4. **Create a system configuration profile that defines the contents of the Kerberos configuration file.**

   This example creates a profile by running the `kclient` command interactively. Alternatively, you could invoke the command using command-line options or using an input profile. For more information see the `kclient`(8) man page.

   In this example, the KDC is running on an MIT server. To view sample output for a Solaris KDC, see Example 25, “Downloading Existing Keys While Deploying Kerberos Clients,” on page 61. To view sample output for an AD client, see Example 27, “Automatically Joining an Kerberos Client to a MS AD Domain,” on page 62.

```
$ kclient -x /root/krb-sc.xml
Starting client setup
-------------------------------------------------
Is this a client of a non-Solaris KDC ? [y/n]: y
Which type of KDC is the server:
       ms_ad: Microsoft Active Directory
       mit: MIT KDC server
       heimdal: Heimdal KDC server
       shishi: Shishi KDC server
Enter required KDC type: mit
Do you want to use DNS for Kerberos lookups ? [y/n]: n
       No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the master KDCs for the above realm using a comma-separated list: kdc.example.
com
Do you have any slave KDC(s) ? [y/n]: y
Enter a comma-separated list of slave KDC host names: kdc2.example.com
Do you have multiple domains/hosts to map to a realm ? [y/n]: n
       No action performed.
Setting up /root/krb-sc.xml.
```

5. **(Optional) Convert a Kerberos client's binary keytab file into an XML profile.**

   This step is not needed if the keys can be obtained through auto-registration or if the Kerberos client is keyless. The Kerberos client needs to have a keytab file created, which is often done by the KDC administrator when a client is first configured.

```
$ kclient-kt2prof -k ./host1.keytab -p /root/host1.xml
```

6. **Create client profiles to configure the rest of the Kerberos client.**

Because a profile must be used in this procedure, configure as much of the Kerberos client as possible using system configuration profiles.

7. **(Optional) Set the security policy for profiles.**
   If the client profiles include a keytab, you should assign the `require-client-auth` security policy to the service so that only authenticated clients can download their keytab file.

   ```
   $ installadm set-service -p require-client-auth -n krb-sparc
   ```

8. **Associate the client profiles with the install service.**
   Associate the profiles for the Kerberos configuration file, the client keytab file, and any other profiles that you have created to the install service.

   ```
   $ installadm create-profile -n krb-sparc -f /root/krb-sc.xml
   Profile krb-sc.xml added to database.
   $ installadm create-profile -n krb-sparc -f /root/host1.xml -c mac="11:11:11:11:11:11"
   Profile host1.xml added to database.
   ```

9. **Boot the Kerberos client to start the AI process.**

Example 25 Downloading Existing Keys While Deploying Kerberos Clients

Note that using auto-registration only works if the KDC is either an Oracle Solaris KDC or an MS AD. If the KDC is MIT, Heimdal or Shishi, only pre-generated keytab transfer is possible.

In order to use auto-registration to download existing keys, you must first have created an admin principal on the KDC with `c` and `i` administration privileges. In this example, the name of the principal is `download/admin`.

In this example, the KDC is running Oracle Solaris. Also, the keys for the Kerberos client have already been created.

This example shows how to add the `download/admin` principal when you are creating the system configuration profile for the Kerberos configuration file. The `download/admin` principal is a special admin principal that is used to transfer existing keys from the KDC server when the Kerberos client is deployed.

```
$ kclient -x /root/krb-sc.xml
Starting client setup
-------------------------------------------------
Is this a client of a non-Solaris KDC ? [y/n]: n
        No action performed.
Do you want to use DNS for Kerberos lookups ? [y/n]: n
        No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the master KDCs for the above realm using a comma-separated
```

```
list: kdc.example.com
Do you have any slave KDC(s) ? [y/n]: y
Enter a comma-separated list of slave KDC host names: kdc2.example.com
Do you have multiple domains/hosts to map to realm  ? EXAMPLE.COM [y/n]: n
        No action performed.
Should the client automatically join the realm ? [y/n]: y
Enter the krb5 administrative principal to be used: download/admin
Password for download/admin: xxxxxxxx
Do you plan on doing Kerberized nfs ? [y/n]: n
        No action performed.
Is this client a member of a cluster that uses a logical host name ? [y/n]: n
        No action performed.
Do you have multiple DNS domains spanning the Kerberos realm EXAMPLE.COM ? [y/n]: n
        No action performed.
Setting up /root/krb-sc.xml.
```

Example 26 Creating New Keys While Deploying Kerberos Clients

Note that using auto-registration only works if the KDC is either an Oracle Solaris KDC or an MS AD. If the KDC is MIT, Heimdal or Shishi, only pre-generated keytab transfer is possible.

In order to use auto-registration to download new keys, you must first have created an admin principal on the KDC with `a`, `c` and `i` administration privileges. In this example, the name of the principal is `create/admin`.

In this example, the KDC is running Oracle Solaris. This example adds the `create/admin` principal when you are creating the system configuration profile for the Kerberos configuration file. The `create/admin` principal is a special admin principal that is used to transfer new keys from the KDC server when the Kerberos client is deployed. This command includes more options so fewer questions are asked.

```
$ kclient -x /root/krb-sc.xml -R EXAMPLE.COM -a create/admin -d none -m kdc.example.com
Starting client setup
-------------------------------------------------
Do you have multiple domains/hosts to map to realm  ? EXAMPLE.COM [y/n]: n
        No action performed.
Should the client automatically join the realm ? [y/n]: y
Password for create/admin: xxxxxxxx
Setting up /root/krb-sc.xml.
```

Example 27 Automatically Joining an Kerberos Client to a MS AD Domain

In this example, the Kerberos client is joining an AD domain. Use the following command to add the `Administrator` principal when you are creating the system configuration profile for the Kerberos configuration file.

```
$ kclient -x /root/krb-sc.xml
```

```
Starting client setup
-------------------------------------------------
Is this a client of a non-Solaris KDC ? [y/n]: y
Which type of KDC is the server:
        ms_ad: Microsoft Active Directory
        mit: MIT KDC server
        heimdal: Heimdal KDC server
        shishi: Shishi KDC server
Enter required KDC type: ms_ad
Should the client automatically join AD domain ? [y/n]: y
Enter the Kerberos realm: EXAMPLE.COM
Enter the krb5 administrative principal to be used: Administrator
Password for Administrator: xxxxxxxx
Setting up /root/krb-sc.xml.
```

Other Security Related Tasks

These section describes other tasks specific to security for automated installations.

Ensure that your role has the appropriate rights profiles to perform this procedure. See “Using Rights Profiles to Install Oracle Solaris” on page 18.

Modifying the AI Manifest to Install From a Secure IPS Repository

If an AI manifest specifies a publisher that has a secure origin, specify the key and certificates in the `credentials` sub-element of the `publisher` element. See the Software section of the `ai_manifest(5)` man page for details. You can specify an SSL key and certificate in attributes of the `image` element, but this key and certificate apply only to the first publisher specified in the manifest. If keys and certificates are specified both in an `image` element and in a `credentials` element, the credentials specified in the `credentials` element are used. Consider locating key and certificate files in a user-specified directory on the AI web server. See “Configuring the AI Server's Web Server Files Directory” on page 30 for information.

Disabling and Enabling Security

This section describes the options you can use to disable security requirements without deleting the security configuration, and then re-enable security requirements using the previously configured server and client authentication settings.

Security is enabled by default. While security is disabled, no credentials are issued to clients and no credentials are required from clients. While security is disabled, no HTTPS network protection is provided for any of the AI files served to a client. User-specified secure files served by the AI web server (as described in “Configuring the AI Server's Web Server Files Directory” on page 30) are not accessible while security is disabled.

While security is disabled, you can continue to configure security. Any changes are effective when security is re-enabled.

Use the following command to disable security enforcement server-wide:

```
$ installadm set-server -S
Refreshing web server.
Automated Installer security has been disabled.
```

Use caution when disabling security for systems that already have install services configured. The secured install service data will not require authentication to access, and non-authenticated clients will be able to install Oracle Solaris through AI.

Use the following command to re-enable security enforcement after security enforcement has been disabled with `set-security --disable`:

```
$ installadm set-security -s
Configuring web server security.
Refreshing web server.
Warning: client 02:00:00:00:00:00 of service solaris11_4-i386
is required to have credentials but has none.
Automated Installer security has been enabled.
```

Deleting Credentials

Use the `installadm` command to delete security credentials. The `set-server`, `set-service`, and `set-client` subcommands can be used to delete security credentials.

Security credentials are also removed when you run the `delete-client` or `delete-service` subcommands. The `delete-client` command removes all client-specific credentials. The `delete-service` subcommand removes all service-specific credentials as well as any client-specific credentials for all clients of that service and any alias services.

Caution - Deleted credentials cannot be recovered, and the TLS security protocol cannot function without server credentials. AI security will be disabled prior to deleting the server credentials.

EXAMPLE 28 Deleting Credentials for One Client

This example deletes the private key and certificate, any CA certificate, and any OBP keys that were assigned to the client by using a MAC address.

```
$ installadm set-client -e mac-addr -x
```

EXAMPLE 29 Deleting a CA Certificate

This example deletes the specified CA certificate for all clients that use that CA certificate. The value of the --hash option argument is the hash value of the certificate's X.509 subject, as displayed by the `list` subcommand and shown in "Displaying Client Security Information" on page 57. Any clients that are using the specified CA certificate are counted and displayed along with a prompt to confirm you want to continue.

```
$ installadm set-client -x --hash b99588cf
  Identifier hash: b99588cf
  Subject: /C=CZ/O=Oracle Czech s.r.o./OU=install/CN=genca
  Issuer:  /C=CZ/O=Oracle Czech s.r.o./OU=install/CN=genca
  Valid from Apr 27 13:12:27 2012 GMT to Apr 27 13:12:27 2015 GMT
This CA has the following uses:
        WARNING: this is the server CA certificate
Deleting this Certificate Authority certificate can prevent
    credentials from validating.
Do you want to delete this Certificate Authority certificate [y|N]: y
Deleting all references to Certificate Authority with hash value b99588cf
```

Caution -In this example, all instances of this CA certificate are deleted for all clients that use it; the affected clients can no longer be authenticated. Once the specified CA certificate is used to generate certificates, the `installadm` command can no longer generate certificates.

EXAMPLE 30 Deleting AI Server Security Credentials

This example deletes the AI server's private key and certificate, any CA certificate, and the OBP keys for server authentication only:

```
$ installadm set-server -x
```

♦ ♦ ♦ CHAPTER 6

Running AI on Clients by Using an AI Server

This chapter provides client configuration information preparatory to installation. It covers the following topics:

- "Preparing Clients for Installation (Task Map)"
- "Registering Clients With Install Services"
- "Specifying the VLAN for DHCP and AI Services" on page 69
- "Defining Boot Disks on Clients"
- "Starting Automatic Installation on Clients"
- "Performing an Automated Installation: Sample Scenario"

This chapter assumes that you have completed the necessary configurations on the AI server. See Chapter 4, "Setting Up the AI Server".

Before starting any installations on clients, make sure that the systems meet the requirements to support the installation. See "System Requirements for OS Installations" on page 13.

Preparing Clients for Installation (Task Map)

The following task map lists preparations you might need to make before you start the automated installation. Unless specified otherwise, perform these tasks on the client systems.

Task	Description	For More Information
Assign clients to install services.	An optional step to be used in complex AI setups. Ensures that a client uses the correct install service for the installation of the operating system.	"Registering Clients With Install Services"
Specify the VLAN.	If the network uses VLANs, you might need to specify which	"Specifying the VLAN for DHCP and AI Services"

Task	Description	For More Information
	VLAN to use to reach the DHCP and AI services.	
SPARC: Select a boot disk.	An optional task that specifies a boot disk other than the selection made by the installation process.	"How To Set the Boot Disk From OBP"
SPARC: Check encryption and firmware keys.	Ensures that in SPARC clients, the hash and firmware keys match the keys that have been generated on the AI server.	"Configuring WAN Boot Security for SPARC Clients"
Enable installation monitoring on the client.	An optional task that sets clients to accept network access to enable installation monitoring.	"Setting Secure Shell for Remote Monitoring"

Registering Clients With Install Services

In a simple AI setup, you would not need to create clients for install services. When you create an architecture-specific install service for the first time on the AI server, a corresponding default service is also created. See "Creating and Configuring an Install Service" on page 33. Through the default service, any client can have the operating system installed on it using AI, provided that the client system's has a matching architecture or platform.

However, in complex AI setups, you would need to specifically assign a system client to an install service in order for the client to be serviced. For example, suppose that one of an AI service's manifest has install instructions that you want to implement on a client system. You would register that client system with the install service. Additionally, you would also specify manifest criteria to ensure that the correct install service and the manifest are used when the client boots from the network. For more information about special install instructions in customized manifests, see *Customizing Automated Installations With Manifests and Profiles*.

You can register as many clients for a service as needed. These clients are identified by the install service through the clients' MAC addresses.

To create AI clients, use the `create-client` subcommand. For example:

```
aiserver$ installadm create-client -n x86service -e aa:bb:cc:dd:ee:ff
```

Display the results as follows:

```
aiserver$ installadm list -c
Service Name  Client Address     Arch  Secure  Custom Args  Custom Grub
x86service    aa:bb:cc:dd:ee:ff  i386  no      no           no
```

Specifying the VLAN for DHCP and AI Services

If the network uses VLANs, you might need to specify on the AI client which VLAN to use to reach the DHCP and AI services.

By default, network boot is performed by using untagged Ethernet frames, relying on the switch to provide automatic tagging. Some sun4v SPARC systems can network boot over a VLAN that requires tagging by the client. To enable network boot over a VLAN that requires tagging by the client, specify the VLAN ID as a modifier to the `net` path.

The following example shows the command to network boot on VLAN 816:

```
boot net:vlan=816
```

When you specify DHCP configuration in the `boot` command, the `vlan` modifier must precede the `dhcp` modifier on the command line.

The following example shows the command to specify DHCP configuration with a tagged VLAN:

```
boot net:vlan=816,dhcp
```

To verify whether tagged VLANs are used on the AI DHCP server, use the `dladm show-vlan` command.

SPARC: Defining Boot Disks on Clients

The manifest typically contains the boot disk definition. If the definition is absent, set the `boot-device` OBP parameter to specify where to install the OS. Otherwise, the installer uses the first disk available that has sufficient space for the OS.

▼ How To Set the Boot Disk From OBP

Before You Begin Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. **Bring the system to the `ok` PROM prompt.**

   ```
   # init 0
   ```

2. **List the devices on the system.**

```
ok devalias
...
disk1                    /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@w202400a0b836a3b9,3
disk0                    /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@w202400a0b836a3b9,1
disk                     /pci@304/pci@2/usb@0/storage@1/disk@0,0
```

3. **Set the `boot-device` parameter to the appropriate disk.**

   For example:

   ```
   ok setenv boot-device /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@x202400a0b836a3b39,1
   ```

4. **(Optional) Verify the boot device.**

   ```
   ok printenv boot-device
   boot-device = /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@x202400a0b836a3b39,1
   ```

Starting Automatic Installation on Clients

To start the installation on a client, boot the client from the network. When connections with the AI server are established, the installation begins.

▼ How to Start the Installation on SPARC Clients

Before You Begin You must complete the necessary configurations on the AI server. See Chapter 4, “Setting Up the AI Server”.

Ensure that your role has the appropriate rights profiles to perform this procedure. See “Using Rights Profiles to Install Oracle Solaris” on page 18.

1. **Access the system's OBP prompt.**

   ```
   $ init 0
   ```

2. **Choose one of the following commands to boot the client:**

   - If you are using DHCP:

     ```
     ok boot net:dhcp - install
     ```

   - If you want to enable or disable remote access for this particular client:

     ```
     ok boot net:dhcp - livessh=enable|disable
     ```

Note - Remote access must be enabled on the server. See "Setting Secure Shell for Remote Monitoring" on page 32.

- If you want the client to use mDNS to locate the correct install server among multiple servers:

```
ok boot net:dhcp - install mdns
```

- If you are not using DHCP:

```
ok setenv network-boot-arguments host-ip=client-ip, \
   router-ip=router-ip,subnet-mask=subnet-mask,hostname=hostname, \
   file=wanboot-cgi-file
```

The `network-boot-arguments` variable is set persistently in OBP. Then boot the client:

```
ok boot net - install
```

Note - When you use the `network-boot-arguments` variable, the SPARC client does not have DNS configuration information. Ensure that the AI manifest used with this client specifies an IP address instead of a host name for the location of the IPS package repository, and for any other URI in the manifest.

At the completion of the installation, the systems would reboot if `auto_reboot` is set to `true` in the clients' manifest. Otherwise, you would be prompted to reboot the client manually.

3. **(Optional) To remotely access the client, use the `ssh` command to log in.**

   For first-time client access, use `jack` as both user name and password to log in.

   The progress of the installation is recorded in the `/system/volatile/install_log` file.

How to Start the Installation on x86 Clients

Before You Begin You must complete the necessary configurations on the AI server. See Chapter 4, "Setting Up the AI Server".

Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. **Boot the system.**

2. **While the system is booting, press the appropriate function key to access the BIOS screen.**

   For example, some systems use F12.

3. **Configure the boot device information to boot the system from the network, then continue booting.**

   The GRUB menu displays the options of using either the text installer or the automated installer to install the operating system.

4. **Choose to start the automated installation.**

   At the completion of the installation, the systems would reboot if `auto_reboot` is set to `true` in the clients' manifest. Otherwise, you would be prompted to reboot the client manually.

5. **(Optional) To remotely access the client, use the `ssh` command to log in.**

   For first-time client access, use `jack` as both user name and password to log in.

   Note - Remote access must be enabled on the server. See "Setting Secure Shell for Remote Monitoring" on page 32.

   The progress of the installation is recorded in the `/system/volatile/install_log` file.

Performing an Automated Installation: Sample Scenario

This section puts together the steps to run automated installations, beginning with AI server configuration.

Note - For more information about customizing installations with AI, see *Customizing Automated Installations With Manifests and Profiles*. A good example of a complex AI setup is provided in `https://blogs.oracle.com/solaris/customize-network-configuration-with-solaris-automated-installs-v2`.

The scenario has the following assumptions:

- The environment consists of x86 systems only.
- The system hosting the AI server is running Oracle Solaris 11.4 that is fully configured.
- Installation has two target clients: Client A and Client B.
- Both clients use a common manifest with the following parameter definitions:
  - At the end of the installation, reboot automatically.

    - Add the `desktop` package when installing the OS.
- Each client uses its own configuration profile with client-specific time zone settings.

Note - Output in the examples is truncated for brevity. Edits to XML manifest and configuration files are in bold.

Set up the AI server to manage DHCP.

```
$ installadm set-server -m -i 203.0.113.10 -c 20
    Process takes a while.
$ svcs -a | grep ipv4
...
online     7:18:18 svc:/network/dhcp/server:ipv4
```

Create the install service.

```
$ installadm create-service -n s11_4-i386
...
  0% : Created Service s11_4-i386
  0% : Refreshing SMF service svc:/system/install/server:default

$ installadm list
Service Name     Status Arch Type Secure Alias Aliases Clients Profiles Manifests
------------     ------ ---- ---- ------ ----- ------- ------- -------- ---------
s11_4-i386       on     i386 pkg  no     no    1       0       0        1
```

Create an XML manifest file.

For example, create and edit `/var/tmp/x86manifest.xml` with the following settings:

```
  <ai_instance name="x86manifest" auto_reboot="true">
...
<sofware_data action="install">
   other-packages
   <name>pkg:/group/system/solaris-desktop</name>
</software_data>
```

Create the manifest instance based on the XML file.

```
$ installadm create-manifest -m x86manifest \
   -f /var/tmp/x86manifest.xml -n s11_4-i386
Created Manifest x86manifest

$ installadm list -m
Service Name           Manifest Name      Type      Status    Criteria
------------           -------------      ----      ------    --------
s11_4-i386             x86manifest        xml       active    none
```

Create two XML files as profiles, each with a specific time zone setting.

- `/var/tmp/profileA.xml`

```
<service_bundle type="profile" name="profileA">
...
  <service name="system/timezone" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="timezone" type="application">
        <propval name="localtime" type="astring" value="US/Pacific"/>
      </property_group>
    </instance>
  </service>
```

- `/var/tmp/profileB.xml`

```
<service_bundle type="profile" name="profileB">
...
  <service name="system/timezone" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="timezone" type="application">
        <propval name="localtime" type="astring" value="Asia/Tokyo"/>
      </property_group>
    </instance>
  </service>
```

Create corresponding profile instances for the service with MAC addresses as criteria.

```
$ installadm create-profile -p profileA -f /var/tmp/profileA.xml \
   -c aa:bb:cc:dd:ee:ff -n s11_4-i386
Created Profile profileA
$ installadm create-profile -p profileB -f /var/tmp/profileB.xml \
   -c 00:aa:11:bb:22:cc -n s11_4-i386
Created Profile profileB

$ installadm list -p
Service Name   Profile Name   Environment   Criteria
------------   ------------   -----------   --------
s11_4-i386     profileA       system        mac = aa:bb:cc:dd:ee:ff
               profileB       system        mac = 00:aa:11:bb:22:cc
```

Create the clients for the service and specify their MAC addresses for identification.

```
$ installadm create-client -e aa:bb:cc:dd:ee:ff -n s11_4-i386
adding host entry for aa:bb:cc:dd:ee:ff to local DHCP configuration.
Created Client aa:bb:cc:dd:ee:ff

$ installadm create-client -e 00:aa:11:bb:22:cc -n s11_4-i386
```

```
adding host entry for 00:aa:11:bb:22:cc to local DHCP configuration.
Created Client 00:aa:11:bb:22:cc

$ installadm list -c
Service Name   Client Address      Arch   Secure   Custom   Custom   Status
                                                   Args     Grub
------------   --------------      ----   ------   ------   ------   ------
s11_4-i386     aa:bb:cc:dd:ee:ff   i386   no       no       no       Not started
               00:aa:11:bb:22:cc   i386   no       no       no       Not started
```

Boot each client.

During the boot process, press the function key to access the BIOS screen and then select to boot from the network. Continue booting and from the GRUB menu, select to run the OS automatic install.

♦ ♦ ♦ CHAPTER 7

Installing and Configuring Zones

This chapter describes how to include non-global zones as part of an Oracle Solaris installation. It covers the following topics:

- "About Zone Installations Using AI"
- "Preparing Automated Installation of Zones"
- "How to Include a Non-Global Zone in an Automatic Installation"

About Zone Installations Using AI

AI supports non-global zone installations. The operation occurs at the completion of the global zone OS installation. After the client system reboots, the `svc:/system/zones-install:default` service configures and installs the non-global zones.

Kernel zones can also be automatically installed provided that they have the same OS version as the global zone. Otherwise, you would need to archive the kernel zones first in order to be included in the installation. To create unified archives, see *Using Unified Archives for System Recovery and Cloning in Oracle Solaris 11.4*.

Oracle Solaris 10 zones are not supported in AI. For these, you would need to use a first-boot script. For more information about scripts, see Chapter 6, "Running a Custom Script During First Boot" in *Customizing Automated Installations With Manifests and Profiles*.

Zones are not installed if any of the following errors occurs:

- A zone configuration 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 datasets are not configured in the global zone.

Preparing Automated Installation of Zones

Several preparatory steps are required to successfully include non-global zones in an AI installation.

Creating Zone Configuration Files

AI requires the configuration files of the non-global zones that you want to install on the client.

Zone configuration files are different from system configuration profiles that are typically used with AI. Zone configuration files contain information specific to their corresponding non-global zones. Therefore, you must create non-global zones first on a working system. Then you extract zone information into zone configuration files.

For instructions to create and configure non-global zones, see Chapter 2, "Setting Up a Non-Global Zone" in *Creating and Using Oracle Solaris Zones*.

Instructions for the creation of zone configuration files for AI are described in "How to Include a Non-Global Zone in an Automatic Installation" on page 80.

Preparing Zone Specific Files for AI

Just like with system or global zone installations, AI also uses manifests and system configuration profiles to install non-global zones.

Non-Global Zone AI Manifest

For its manifest, a non-global zone installation uses `/usr/share/auto_install/manifest/zone_default.xml` or customized copies of that file. The parameters in this file are more limited than in manifests for global zone installations. Do not use `default.xml` or copies of it as templates for non-global zone manifests.

The `zone_default.xml` file has 3 sections with their corresponding elements:

- `target` section
  - `target/logical/zpool` – Specify a name. Only one pool can be defined.

- `target/logical/zpool/filesystem` entries – Include file systems and their mount points.
- `target/logical/zpool/be` entries – Specify a boot environment to use.

- `software_type` section
  - By default, the software type is IPS and is the only supported type. Do not change this value in the manifest.
  - `software_type/destination/image` – Set to `true` only the locales you want to install. By default, all locales are set to `true`, which means that all locales will be installed.
- `software_data` section

  The elements and attributes already have default definitions.
  - `software_data action` – By default, the action is set to perform an installation.
  - `software_data/name` – By default, AI would install the `solaris-small-server` package, which contains typical tools and device drivers for non-global zones.

To use a manifest for a non-global zone AI installation, create a copy of `/usr/share/auto_install/manifest/zone_default.xml`. Then, customize the manifest by editing specific parameters in the file.

In the following manifest example, the elements and attributes relevant to non-global zone installations are displayed in bold. The `software_data` section has been customized so that AI installs the zone with the `solaris-large-server` package and to install only the English locales.

```
auto_install>
   <ai_instance name="myzone">
      <target>
         <logical>
            <zpool name="rpool">
...
               <filesystem name="export" mountpoint="/export"/>
               <filesystem name="export/home"/>
               <be name="solaris">
                  <options>
                     <option name="compression" value="on"/>
                  </options>
               </be>
            </zpool>
         </logical>
      </target>

      <software type="IPS">
         <destination>
            <image>
               <!-- Specify locales to install -->
```

```
                <facet set="false">facet.locale.*</facet>
                <facet set="false">facet.locale.de</facet>
                <facet set="false">facet.locale.de_DE</facet>
                <facet set="true">facet.locale.en</facet>     English locales
                <facet set="true">facet.locale.en_US</facet>
                <facet set="false">facet.locale.es</facet>
                <facet set="false">facet.locale.es_ES</facet>
                <facet set="false">facet.locale.fr</facet>
                <facet set="false">facet.locale.fr_FR</facet>
                <facet set="false">facet.locale.it</facet>
                <facet set="false">facet.locale.it_IT</facet>
                <facet set="false">facet.locale.ja</facet>
                <facet set="false">facet.locale.ja_*</facet>
                <facet set="false">facet.locale.ko</facet>
                <facet set="false">facet.locale.ko_*</facet>
                <facet set="false">facet.locale.pt</facet>
                <facet set="false">facet.locale.pt_BR</facet>
                <facet set="false">facet.locale.zh</facet>
                <facet set="false">facet.locale.zh_CN</facet>
                <facet set="false">facet.locale.zh_TW</facet>
             </image>
          </destination>
          <software_data action="install">
             <name>pkg:/group/system/solaris-large-server</name>
          </software_data>
       </software>
    </ai_instance>
</auto_install>
```

Non-Global Zone System Configuration Profiles

Tools and templates for creating configuration profiles are the same for both global and non-global zone installations. No configuration profile template specific only to zones exists.

You can create multiple profiles as needed. For instructions, see Chapter 3, “Working With System Configuration Profiles” in *Customizing Automated Installations With Manifests and Profiles*.

▼ How to Include a Non-Global Zone in an Automatic Installation

In this procedure, steps performed on the AI server are specifically indicated.

Before You Begin The non-global zone must already be created and properly configured on a separate working system.

Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. **On the system where your current non-global zone exists, create its configuration file.**

   a. **(Optional) Display the non-global zone's live configuration.**

      ```
      $ zonecfg -z zone-name -r info
      ```

   b. **Export the live configuration.**

      Use the following command syntax to create zone configuration files.

      ```
      $ zonecfg -a zone-name export -f filename.cfg
      ```

   c. **Store the configuration file at a location that is network accessible.**

      You can also store the configuration file directly on the AI server itself.

2. **On the non-global zone manifest, define installation instructions.**

   Use a copy of the template `/usr/share/auto_install/manifest/zone_default.xml`.

3. **Associate the non-global zone manifest with the install service of the global zone.**

   Suppose that your non-global zone manifest file is `/tmp/zmanifest.xml`, the manifest instance name is `zmanifest`, and the install service is `solaris11_4-sparc`. To use the manifest to install `zone1` and `zone2`, you would issue the following command:

   ```
   aiserver$ installadm create-manifest -n solaris11_4-sparc -f /tmp/zmanifest.xml \
      -m zmanifest -c zonename="zone1 zone2"
   ```

4. **(Optional) Prepare the configuration profile for the zone.**

   a. **Create the XML file that contains the zone's configuration profile.**

      See "Non-Global Zone System Configuration Profiles" on page 80 and Chapter 3, "Working With System Configuration Profiles" in *Customizing Automated Installations With Manifests and Profiles*.

   b. **Associate the profiles with the install service of the global zone.**

      Suppose that you created two profiles for the install service `solaris11_4-sparc`:

- `/tmp/z1profile.xml` with instance name `z1profile` to be used for `zone1` and `zone2`
- `/tmp/z2profile.xml` with instance name `z2profile` to be used further for `zone2`

You would issue the following commands:

```
aiserver$ installadm create-profile -n solaris11_4-sparc -f /tmp/z1profile.xml \
   -p z1profile -c zonename="zone1 zone2"

aiserver$ installadm create-profile -n solaris11_4-sparc -f /tmp/z2profile.xml \
   -p z2profile -c zonename="zone2"
```

5. Configure the global zone manifest to install the non-global zones.

Assume that the global zone manifest is `line1-netra2000.xml`. Configure that file's `configuration` element by choosing from one of the following methods, depending on what zone files you are using:

- **If you are using zone configuration files (`*.cfg`)**

  Specify the zone configuration files on the `configuration` element. Suppose that the zone configuration file is `zconfig.cfg` in `system1`. The file is used to install both `zone1` and `zone2`. You would add the following lines to the bottom of the file:

```
...
  <configuration type="zone" name="zone1"
    source="http://system1/tmp/zconfig.cfg"/>
  <configuration type="zone" name="zone2"
    source="http://system1/tmp/zconfig.cfg"/>

  </ai_instance>
</auto_install>
```

- **If you are using a zone archive file**

  Specify the zone archive on the `configuration` element. Suppose that the archive file `zsystem.uar` contains the zone configuration of `orig-zone`. You want to apply the archived configuration to both `zone1` and `zone2`. You would add the following lines to the bottom of the file:

```
...
  <configuration type="zone" name="zone1"
    source="archive:orig-zone" archive="http://system1/archives/zsystem.uar"/>
  <configuration type="zone" name="zone2"
    source="archive:orig-zone" archive="http://system1/archives/zsystem.uar"/>

  </ai_instance>
</auto_install>
```

6. **(Optional) Display the manifests and configuration profiles associated with the service.**

   The install service name is `solaris11_4-sparc`.

```
aiserver$ installadm list -n solaris11_4-sparc -m -p

Service Name      Manifest Name   Type    Status  Criteria
------------      -------------   ----    ------  --------
solaris11_4-sparc line1-netra2000 xml     active  mac = 00:14:4F:2D:7A:DC
                  zmanifest       xml     active  zonename = zone1,zone2
                  orig_default    derived default none

Service Name      Profile Name Environment Criteria
------------      ------------ ----------- --------
solaris11_4-sparc z1profile    system      zonename = zone1,zone2
                  z2profile    system      zonename = zone2
```

7. **Start the installation by booting the client from the network.**

   See Chapter 6, “Running AI on Clients by Using an AI Server”.

8. **(Optional) Trace the progress of a non-global zone installation.**

   You can either monitor the `system/zones-install` service or check the output of the `zoneadm list -cv` command.

♦ ♦ ♦ CHAPTER 8

Automated Installations That Boot From Media

You can initiate an automated installation by booting from a removable media instead of over the network. This chapter covers the following topics:

- "Overview of Installation Using AI Media"
- "Installing Using AI Media"

Overview of Installation Using AI Media

Using an AI media enables you to do the following:

- Install Oracle Solaris on the system that will serve as your AI server.
- Install on a SPARC system that does not have WAN boot capability.
- Troubleshoot an ailing system. You can boot the system from the removable media and then inspect the installed system and run diagnostics.

Installing Using AI Media

Installations using AI media installs the OS only on the system that contains the media. An AI manifest is automatically supplied with the downloaded image.

System Requirements

Both SPARC and x86 systems must meet the following requirements:

- Memory and disk space requirements as specified in the *Oracle Solaris 11.4 Release Notes*
- Network Access to access an IPS repository of the packages to be installed as well as the HTTP server where a custom manifest, if used, is stored.

SPARC: How to Create a Persistent Device Alias for a USB Flash Drive

With a persistent device alias, you would not need to re-create the alias provided that you reuse the same port for the USB drive.

Before You Begin Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. **Shut down the system and leave it at the boot prompt.**

2. **Display the available disks on the system.**

```
ok show-disks
```

3. **Select the device to use for the USB drive.**

This example shows the selection of the second device.

```
{0} ok show-disks
      a) /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@3/disk
      b) /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk
      c) /pci@400/pci@0/pci@1/scsi@0/disk
      d) /iscsi-hba/disk
      q) NO SELECTION

    Enter Selection, q to quit: b
      /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk has been selected.
      Type ^Y ( Control-Y ) to insert it in the command line.
      e.g. ok nvalias mydev
              for creating devalias mydev for
      /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk
```

4. **Set an alias for the USB flash drive.**

```
{0} ok nvalias usbdrive ^Y
```

How to Install Using AI Media

Before You Begin Ensure that your role has the appropriate rights profiles to perform this procedure. See "Using Rights Profiles to Install Oracle Solaris" on page 18.

1. **Download the AI boot image.**

a. **Go to `https://www.oracle.com/solaris/solaris11/downloads/solaris-downloads.html`.**

b. **Select** *Installation from CD/DVD or USB***.**

c. **Under the Automated Installer Boot Image heading, download either the ISO image or the USB image.**

   Make sure to download the image that corresponds to the client platform on which the OS will be installed.

2. **Review the default AI manifest (`default.xml`).**

   a. **View the contents of the manifest copy.**

      In the following example, the downloaded image is mounted so you can copy the manifest to a target file for viewing.

```
$ /usr/sbin/mount -o ro -F hsfs /tmp/sol-11_4-20-ai-x86.iso /mnt
$ cp /mnt/auto_install/manifest/default.xml /home/username/custom.xml
$ umount /mnt
```

   b. **If necessary, add instructions to the copied manifest.**

      See Chapter 3, “Working With System Configuration Profiles” in *Customizing Automated Installations With Manifests and Profiles*.

   c. **Store the customized manifest on an HTTP server.**

3. **Perform one of the following:**

   - **If you downloaded the ISO image, ensure that it is on an accessible location.**

   - **If you downloaded the USB image, copy the image to a USB media.**

     If the system does not have any Oracle Solaris installed, use the `dd` command. For example:

```
# dd bs=16k conv=sync if=image-path of=/dev/rdsk/disk
```

     If the system with the downloaded image has an existing Oracle Solaris installation, you can use the `usbcopy` command instead of `ddu`, provided that the `pkg:/install/distribution-constructor` package has also been installed.

4. **Boot from the device that contains the boot image.**

Choose one of the following boot methods depending on your system's platform.

- **Booting a SPARC system**

  a. **From the OBP prompt, use the `boot` command with appropriate options.**

  To boot using the default AI manifest:

  ```
  ok> boot media - install
  ```

  To boot using a customized manifest:

  ```
  ok> boot media - install aimanifest=prompt
  ```

  At the prompt, specify the manifest's path in the HTTP server, such as `http://example.com/custom.xml`.

  To boot without installing for troubleshooting purposes:

  ```
  ok> boot media
  ```

- **Booting an x86 system**

  a. **From the GRUB menu, decide whether to install with the default AI manifest or a customized manifest.**

  b. **For either method, select an option depending on where you want installation output to be displayed:**

  Select `ttya` to send output to COM1 or `ttyb` to send output to COM2.

  c. **To boot without installing, do the following:**

  i **Type `e` to edit boot arguments.**

  ii **On the `$multiboot` line, remove the `install=true` statement, if present.**

The installation begins. After the operation, the SCI Tool prompts you for added configuration information.

5. **At the SCI Tool prompt, manually provide configuration information on the SCI Tool panels.**

   See Appendix A, “Text Installer Panels,” in *Manually Installing an Oracle Solaris 11.4 System*.

Viewing the Installation Log Files

When the automated installation is complete, the output states whether the installation succeeded or failed.

- If the installation failed, you can review the installation log at `/system/volatile/install_log`.
- If the installation succeeded, you can find the log at `/system/volatile/install_log` before you reboot the system or at `/var/log/install/install_log` after you reboot.

♦ ♦ ♦ CHAPTER 9

Troubleshooting Automated Installations

This chapter discusses some possible failures and how to recover. It covers the following topics:

General Troubleshooting Methods

This section covers methods you can use to troubleshoot installation problems.

Check the Installation Logs and Instructions

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

The AI manifest that was used for this client is in `/system/volatile/ai.xml`. The system configuration profiles that were used for this client are in `/system/volatile/profile/*`.

You can also check the AI server's webserver log in `/var/ai/image-server/logs/*`. This log is useful if the client is receiving an unexpected or incorrect AI manifest or system configuration profile, or when the client can not download files after the initial boot phase.

Check DNS

Check whether DNS is configured on your client by verifying that a non-empty `/etc/resolv.conf` file exists.

If `/etc/resolv.conf` does not exist or is empty, check that your DHCP server is providing DNS server information to the client:

```
# /sbin/dhcpinfo DNSserv
```

If this 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.

If an `/etc/resolv.conf` file exists and is properly configured, check for the following possible problems and contact your system administrator for resolution:

- The DNS server might not be resolving your IPS repository server name.
- No default route to reach the DNS server exists.

Client Boot Errors

This section describes errors that might occur during booting. The errors are further sorted according to platforms.

Boot Disk Not Found

If the boot disk is not found during an automated installation, verify the book disk and modify the AI manifest.

1. Select the boot device explicitly in SPARC OBP or in the x86 BIOS.
2. Reboot the system.
3. Log in to the system being installed.
4. Identify the device to be used during the installation. The device is can be identified by the SYS/HDD* receptacle name or the CTD disk name as displayed in the `format` command.
5. Modify the `/system/volatile/ai.xml` manifest and replace the "boot_disk" value. For example:

   ```
   <disk_keyword key="SYS/HDD1" name_type="receptacle"/>
   ```

```
<disk_keyword key="c0t5000CCA012B2A254d0" name_type="ctd"/>
```

6. Refresh the installation service.

```
# svcadm clear auto-installer
```

Related to this issue, when you are installing in UEFI mode, the boot disk cannot be located even if a disk is specified in the AI manifest's target `boot_disk` section. This limitation in UEFI installation means that if nothing is installed on the system yet, you cannot use the `boot_disk` section in the manifest to identify a boot disk. As a workaround, use a derived manifest specify an alternative installation target disk.

SPARC Network Booting Errors and Possible Causes

This section describes errors or problems you might see when booting a SPARC client over the network and possible causes:

Timed out Waiting for BOOTP/DHCP Reply

If a DHCP server is not responding to a SPARC client's request, the following messages display:

```
...
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
   Timed out waiting for BOOTP/DHCP reply
   Timed out waiting for BOOTP/DHCP reply
```

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.

Boot Load Failed

If the 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.

```
Rebooting with command: boot net:dhcp - install
   Boot device: /pci@7c0/pci@0/network@4:dhcp  File and args:
   1000 Mbps FDX Link up
   HTTP: Bad Response: 500 Internal Server Error
   Evaluating:

   Boot load failed
```

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.

Internal Server Error or WAN Boot Alert

After the client has obtained the IP address and initial parameters to start downloading the boot archive, the client might be unable to find or download the `boot_archive`.

- If the client cannot find the `boot_archive`, the following error is displayed:

```
Rebooting with command: boot net:dhcp - install
      Boot device: /pci@7c0/pci@0/network@4:dhcp  File and args:
      1000 Mbps FDX Link up
      <time unavailable> wanboot info: WAN boot messages->console
      <time unavailable> wanboot info: Starting DHCP configuration
      <time unavailable> wanboot info: DHCP configuration succeeded
      <time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
      <time unavailable> wanboot info: wanbootfs: Download complete
      Mon Aug  5 20:46:43 wanboot alert: miniinfo: Request returned code 500
      Mon Aug  5 20:46:44 wanboot alert: Internal Server Error \
```

```
(root filesystem image missing)
```

- If the client finds the `boot_archive` file but cannot access the file, then the following error is displayed:

```
Rebooting with command: boot net:dhcp - install
      Boot device: /pci@7c0/pci@0/network@4:dhcp  File and args:
      1000 Mbps FDX Link up
      <time unavailable> wanboot info: WAN boot messages->console
      <time unavailable> wanboot info: Starting DHCP configuration
      <time unavailable> wanboot info: DHCP configuration succeeded
      <time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
      <time unavailable> wanboot info: wanbootfs: Download complete
      Mon Aug  5 20:53:02 wanboot alert: miniroot: Request returned code 403
      Mon Aug  5 20:53:03 wanboot alert: Forbidden
```

For both of these problems, fix the `boot_archive` file configured for this client. Check the pathname and permissions of the `boot_archive` at `$IMAGE/boot/boot_archive`.

ERROR 403: Forbidden Or ERROR 404: Not Found

The messages `ERROR 403: Forbidden` and `ERROR 404: Not Found` are displayed if the 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. For example, in the following output on a SPARC client, the `solaris.zlib` file does not exist or is not accessible at the specified location:

```
<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
Mon May  5 18:57:36 wanboot progress: miniroot: Read 235737 of 235737 kB (100%)
Mon May  5 18:57:36 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version 11.4 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2015-05-05 18:52:30--  http://203.0.113.67:5555/export/auto_install/11_4_sparc/
solaris.zlib
Connecting to 203.0.113.67:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2015-05-05 18:52:30 ERROR 404: Not Found.
```

```
Could not obtain http://203.0.113.67:5555/export/auto_install/11_4_sparc/solaris.zlib
 from install server
Please verify that the install server is correctly configured and reachable from the AI
 client
```

This problem can be caused by one of the following conditions:

- 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.

Automated Installer Not Started

When installing Oracle Solaris on your client, you need to include the `install` argument when you boot in order to initiate an installation:

```
ok boot net:dhcp - install
```

If you boot without the `install` boot argument, the SPARC client boots into the automated installer boot image but the installation does not start. See “Starting an Automated Installation from the Command Line” on page 108 for instructions about how to start an automated installation from this point.

Invalid HMAC Value

If you see the message `Invalid HMAC value` on the SPARC console shortly after booting a SPARC client, and the system returns to the `ok` prompt, one of the following conditions caused the problem:

- The client is secured by authentication, but you have not set the OBP keys. The solution is to set the OBP keys in the client firmware. For information about authentication, see “Overview of Securing Automated Installations” on page 43.
- The client's install service has a policy requiring client authentication, but no credentials applicable to the client have been assigned. Be sure that there are credentials available for all clients of services with policy `require-client-auth`.

The following steps show how to identify the problem.

1. Verify that security hasn't been disabled for the AI server. Use `installadm list -sv` to see if security is enabled.

2. Verify that security hasn't been disabled for the client's install service. Use `installadm list -vn` *svcname* to see if security not disabled.
3. If the client is using custom credentials, use `installadm list -ve` *macaddr* to obtain the firmware key values.
4. If the client is not a custom client, use `installadm list -vn default-sparc` to see if there are any firmware keys defined for the `default-sparc` service.
5. Check the policy of the client's service with `installadm list -vn` *svcname*.
6. If there are no credentials for the `default-sparc` service, look for default client credentials using the `installadm list -sv` command. If there are default client credentials, then use the firmware keys listed for the default client.
7. If there are no default client credentials, use `installadm list -vn default-sparc` to see if the service policy is set to `require-server-auth`. If so, use the firmware keys listed for the default client in `installadm list -sv`.

x86 Network Booting Errors and Possible Causes

This section describes errors or problems you might see when booting an x86 client over the network and possible causes:

- "No DHCP or ProxyDHCP Offers Were Received" on page 97
- "TFTP Error or System Hangs After GATEWAY Message" on page 98
- "System Hangs After GRUB Menu Entry is Selected" on page 98
- "HTTP Request Sent Results in 403 Forbidden or 404 Not Found" on page 99
- "Automated Installer Not Started" on page 100

X86 System Will Not Boot After Install

The UEFI specification forbids EFI protective MBR partition from being marked as active, but your BIOS implementation requires that at least one hard disk have at least one MBR partition that's marked as bootable/active to boot in BIOS mode. Use the `fdisk` utility to set the partition's status to be "ACTIVE".

No DHCP or ProxyDHCP Offers Were Received

If a DHCP server is not responding to an x86 client's request, you see the following messages:

```
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
```

The timeout message indicates that the client is sending a DHCP request and not getting a response. This issue is probably due to an error in the DHCP configuration. Check whether your client is configured correctly in the DHCP server.

TFTP Error or System Hangs After GATEWAY Message

The DHCP server provides an IP address and a location of the initial boot program as part of the DHCP response.

- If the boot program does not exist, then the client boot cannot proceed. The following message is displayed:

```
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
     CLIENT IP: 203.0.113.0/24   MASK: 198.51.100.0/24    DHCP IP:  203.0.113.0/24
     GATEWAY:203.0.113.0/24     TFTP.
     PXE-T02:    Access Violation
     PXE-E3C: TFTP Error - Access violation
     PXE-MOF: Exiting Intel Boot Agent
```

- If the boot program exists but it is an incorrect program, the client freezes after displaying this message:

```
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
     CLIENT IP: 198.51.100.0/24   MASK: 198.51.100.0/24    DHCP IP:  203.0.113.0/24
     GATEWAY: 203.0.113.0/24
```

System Hangs After GRUB Menu Entry is Selected

If the client is able to do the initial boot but the kernel cannot be booted, the system freezes (or hangs) after you select the entry from the GRUB menu.

On the AI server, check whether the `grub.cfg` file or the `menu.lst` file for this client is pointing to a valid boot archive. The boot directory of the image on the AI server should be loopback-mounted under the `/etc/netboot` directory as shown in this sample excerpt from `df -k` for the image path shown by `installadm list`:

```
Filesystem      1K-blocks      Used Available Use% Mounted on
/export/auto_install/solaris11_4-i386
                 92052473  36629085  55423388  40% /etc/netboot/default-i386
/export/auto_install/solaris11_4-i386
                 92052473  36629085  55423388  40% /etc/netboot/solaris11_4-i386
```

HTTP Request Sent Results in 403 Forbidden or 404 Not Found

On the AI server, if one of the install programs is inaccessible or does not exist in the location specified in the `grub.cfg` file or the `menu.lst` file under `/etc/netboot`, then the client is able to boot, but is not able to download that file. An error message is displayed indicating which file is causing the problem. For example, in the following output on an x86 client, the `solaris.zlib` file does not exist at the specified location:

```
SunOS Release 5.11 Version 11.4 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2015-05-05 20:02:26--  http://203.0.113.0/24/export/auto_install/solaris11_4-i386/
solaris.zlib
Connecting to 203.0.113.66:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2015-05-05 20:02:26 ERROR 404: Not Found.

Could not obtain http://203.0.113.0/24/export/auto_install/solaris11_4-i386/solaris.zlib
 from install server
Please verify that the install server is correctly configured and reachable from the
 client

Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run
```

Check the contents of the target directory that you specified when you ran the `installadm create-service` command.

Automated Installer Not Started

When installing Oracle Solaris on x86 systems for installations that boot over the network, you must select the second entry in the GRUB boot menu to initiate an automated installation. Typically, the menu entries display as follows:

```
Oracle Solaris 11.4 Text Installer and command line
Oracle Solaris 11.4 Automated Install
```

If you selected the first GRUB menu entry or allowed the prompt to time out, the system boots into the automated install boot image but the installation does not start. See “Starting an Automated Installation from the Command Line” on page 108 for instructions about how to start an automated installation from this point.

SPARC and x86 Error Messages

The following errors are common to both SPARC and x86 installations:

mDNS Warning Messages for Multihomed AI Servers

You can ignore warning messages similar to the following example that might appear on the console log:

```
May  5 11:38:41 irperf2 mDNSResponder: [ID 702911 daemon.error] Client
application registered 2 identical instances of service
i86pc_osol_1002-128_is._OSInstall._tcp.local. port 46502.
May  5 11:38:41 irperf2 mDNSResponder: [ID 702911 daemon.error] Client
application registered 3 identical instances of service
i86pc_osol_1002-128_is._OSInstall._tcp.local. port 46502.
...
```

```
May  5 11:38:43 irperf2 mDNSResponder: [ID 702911 daemon.error] Excessive
name conflicts (16) for sun4v_osol_1002-131_is._OSInstall._tcp.local. (SRV);
rate limiting in effect
May  5 11:38:43 irperf2 mDNSResponder: [ID 702911 daemon.error] Excessive
name conflicts (17) for sun4v_osol_1002-131_is._OSInstall._tcp.local. (SRV);
rate limiting in effect
...
```

These messages about multiple identical service instances or about name conflicts might appear in any one of the following cases:

- You have configured the AI server to serve multiple interfaces but not all of its interfaces.
- Your AI server is hosting earlier install service images.

Automated Installer Fails to Install on Systems With High Memory and Low Disk Space Allocation

An AI installation might fail if the system has more physical RAM than disk space. The space allocated to swap and dump devices might reduce the available space for installing the OS. The following error message might be displayed:

```
ImageInsufficentSpace: Insufficient disk space available (8.84 GB) for
estimated need (9.46 GB) for Root filesystem
```

Workaround: Choose one of the following workarounds:

- If not limited by the size of the disk, allocate more space to the slice being used as a virtual device (`vdev`) in the root pool.

  Note - On x86 systems, if necessary, allocate additional space to the `Solaris2` partition.

- Disable the need for a swap volume to be allocated. In the AI manifest, specify the value `true` for the `noswap` attribute in the `<logical>` tag of the `<target>` section. For example:

  ```
  <logical noswap="true">
  </logical>
  ```

- Define the `zpool` and allocate smaller swap and dump sizes in the manifest.

  ```
  <target>
     <disk whole_disk="true" in_zpool="rpool">
        <disk_keyword key="boot_disk"/>
  ```

```
      </disk>
      <logical>
         <zpool name="rpool" root_pool="true">
            <zvol name="swap" use="swap">
               <size val="2gb"/>
            </zvol>
            <zvol name="dump" use="dump">
               <size val="4gb"/>
            </zvol>
            </zpool>
      </logical>
   </target>
```

- Disable allocation of one swap or dump device, and allocate a specific size to the remaining device (dump or swap). The following example shows how to disable swap and add a 4 GB dump size:

```
<target>
   <disk whole_disk="true" in_zpool="rpool">
      <disk_keyword key="boot_disk"/>
   </disk>
   <logical noswap="true">
      <zpool name="rpool" root_pool="true">
         <zvol name="dump" use="dump">
            <size val="4gb"/>
         </zvol>
      </zpool>
   </logical>
</target>
```

For more information about how to edit the AI manifest, refer to the `ai_manifest`(5) man page.

`Automated Installation Failed` Message

If a failure occurs during installation, then the following message is displayed:

```
21:43:34    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
Jul  6 21:43:34 solaris svc.startd[9]: application/auto-installer:default failed
 fatally:
transitioned to maintenance (see 'svcs -xv' for details)
```

IPS Server Not Available

The client needs to reach the IPS package repository defined in the AI manifest in order to install Oracle Solaris. If the client cannot access the package repository, the installation fails and the `application/auto-installer` service transitions to maintenance. The following output is an example of what is displayed on the console:

```
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
15:54:47        generated-transfer-1341-1
15:54:47
15:54:47    Checkpoint execution error:
15:54:47
15:54:47        Framework error: code: 6 reason: Couldn't resolve host 'pkg.example.com'
15:54:47        URL: 'http://pkg.example.com/solaris/release/versions/0/'.
15:54:47
15:54:47    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
Aug 21 15:54:47 line2-v445 svc.startd[8]: 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 Aug 21 15:54:47 UTC 2013
PLATFORM: SUNW,Sun-Fire-V445, CSN: -, HOSTNAME: line2-v445
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: c8a5b809-ece4-4399-9646-d8c64d78aac7
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 https://support.oracle.com/msg/SMF-8000-YX
 for the latest service
procedures and policies regarding this diagnosis.
```

Check the `/system/volatile/install_log` file for messages similar to the following:

```
TransportFailures: Framework error: code: 6 reason: Couldn't resolve host
'pkg.example.com'
URL: 'http://example.com/solaris/versions/0/'
```

```
TransportFailures: Framework error: code: 7 reason: Failed connect to
pkg.example.com:80; Connection refused
URL: 'http://example.com/solaris/versions/0/'

TransportFailures: http protocol error: code: 404 reason: Not Found
URL: 'http://example.com/mysolaris/versions/0/'
```

Depending on which messages you see, try the following possible remedies:

- Try to reach the package server from the failed client, for example, by using `ping`.
- If you are using DNS, check whether DNS is correctly configured on the client. See "Check DNS" on page 92.
- If you are using a local repository, check whether you have made the repository accessible to all clients. See Chapter 3, "Providing Access To Your Repository" in *Creating Package Repositories in Oracle Solaris 11.4*.
- 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.

Package Not Found

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. In the following example, the installer could not find the package `mypkg` in the IPS repository. The following output is an example of what is displayed on the console:

```
14:04:02    Failed Checkpoints:
14:04:02
14:04:02        generated-transfer-1230-1
14:04:02
14:04:02    Checkpoint execution error:
14:04:02
14:04:02        The following pattern(s) did not match any allowable packages.  Try
14:04:02        using a different matching pattern, or refreshing publisher information:
14:04:02
14:04:02                pkg:/mypkg
14:04:02
14:04:02    Automated Installation Failed.  See install log at /system/volatile/
install_log
```

The following output is an example of a portion of the `/system/volatile/install_log` log file:

```
PlanCreationException: The following pattern(s) did not match any allowable packages.
Try using a different matching pattern, or refreshing publisher information:

pkg:/mypkg
```

A related error occurs when you use AI to uninstall packages. The operation fails and the screen displays messages similar to the following example:

```
Failed Checkpoints:
generated-transfer-809-1
Checkpoint execution error:
'pkg:library/python/python-compizconfig-27' matches no installed packages
'pkg:/desktop/compiz/*' matches no installed packages
Automated Installation Fails. See install log at
/system/volatile/install.log
application/auto-installer:default failed fatally: transitioned to
maintenance
```

The AI manifest might contain the following definitions related to the package:

```
<software_data action=install">
<name>pkg:/entire@0.5.11-0.175.3</name>
<name>pkg:/group/system/solaris-large-server</name>
<name>pkg:/group/system/solaris-desktop</name>
<name>pkg:/ncia/nisp/system/firstboot</name>
<name>pkg:/system/device-allocation</name>
</software_data>
<software_data action=uninstall">
<name>pkg:/security/sudo</name>
<name>pkg:/library/python/python-compizconfig-27</name>
<name>pkg:/desktop/compiz/*</name>
<name>pkg:/desktop/compiz</name>
</software_data>
```

In this example, the AI is attempting to remove a package that does not actually exist on the system. Consequently, AI fails.

A case where this error risks occurring is if you uninstall packages using a generic manifest for both SPARC and x86 systems. Both platforms do not entirely share the same packages. Thus, AI would complete on one platform but fail on the other.

A number of workarounds are available to resolve this issue.

- In the case of a missing package from the IPS repository, 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.

- In the case of the uninstalling of non-existing packages, adopt one of the following options before uninstalling:
  - Use the `<reject>` tag in the manifest to list packages to be excluded in the operation.

    Refer to the instructions in the `/usr/share/auto_install/manifest/ai_manifest.xml` sample file for the proper use of this tag.
  - Define your own group package within your own publisher that references the desired packages while omitting the undesired ones.
  - Use a different manifest for each platform.

For an optimal solution, consider using derived manifests that generate manifests specifically based on whatever conditions are desired for particular clients. See Chapter 4, "Using a Script to Customize an Installation" in *Customizing Automated Installations With Manifests and Profiles*.

Boot Errors on Secured Clients

A message similar to the following message when you boot the client means the TLS certificate is not yet valid:

```
SSL3_GET_RECORD:wrong version number - secure HTTPS GET REQUEST to unsecured HTTP port
```

The cause of this problem could be that the system time on the client precedes the time the certificate was generated. Check the system time on the client. See "Overview of Securing Automated Installations" on page 43 for information about how to generate and assign security credentials.

Security-related AI Failures

If you have secured your AI server and clients as described in "Overview of Securing Automated Installations" on page 43, and you are experiencing problems booting or installing those clients, try the following steps to check for authentication errors:

- Check the Apache `access_log` and `error_log` in `/var/ai/image-server/logs/` on the client.
- Log onto the console of the client. Examine the `/tmp/install_log` file and the SMF service logs in `/system/volatile/`.

- If authentication fails after the boot archive loads in the client, when attempting to get image files, AI manifests, or system configuration profiles, you could have a transient networking interruption. Check that the AI server is functioning correctly, and restart the installation.
- Try using the `openssl s_client` command to test the connection:

  ```
  $ openssl s_client -key client-key -cert client-certificate \
  -CAcert server-CA-certificate -connect AI-server-address:port
  ```
- Use the `installadm list -s -v` command to show the enabled or disabled state of security on the AI server. See "Displaying AI Server Security Information" on page 56.
- Check the client's service policy with the `installadm list -v -n` *svcname*`S` command.
- Check assigned credentials against the CA certificates. Use the `-K` and `-C` options with the `installadm list` subcommand to list the assigned keys and certificates. Compare those keys and certificates with the expected keys and certificates using a character comparison utility such as `diff`.
- Make sure the passphrase was removed from `/var/ai/ai-webserver/tls.key/server.key` on the client. X.509 private key files must have any passphrase removed.
- Try using the `wget` command to fetch a file from an AI image, using the appropriate key, certificate, and CA certification, as shown in the following example:

  ```
  $ wget --private-key=client-key --certificate=client-certificate \
  --ca-certificate=server-CA-certificate \
  http://AI-server-address:5555/path-to-file-in-image
  ```

Booting the Installation Environment Without Starting an Installation

Use one of the following methods to boot the installation environment without starting an automated installation.

SPARC client booting over the network

Use the following command to boot a SPARC client over the network without starting an automated installation:

```
ok boot net:dhcp
```

Do not specify the `install` flag as a boot argument.

SPARC client booting from media

Use the following command to boot a SPARC client from media without starting an installation:

```
ok boot cdrom
```

Do not specify the `install` flag as a boot argument.

x86 client booting over the network

For x86 installations that boot over the network, the following GRUB menu displays:

```
Oracle Solaris 11.4 Text Installer and command line
Oracle Solaris 11.4 Automated Install
```

The default entry, "Text Installer and command line", boots the image without starting a hands-free automated installation.

Make sure the entry does not have the `install=true` boot property specified in its kernel line.

x86 client booting from media

If you boot an x86 system from media and do not want to start an installation, edit the GRUB menu and remove the `install=true` boot property from the kernel line of the entry you want to boot.

In general for x86 installations, if the `install=true` boot property is specified in the kernel line of the GRUB entry you are booting from, the installation automatically starts. If you want to boot your x86 based system without initiating an automated installation, check that the GRUB boot entry does not specify the `install=true` boot property. If the property is specified, edit the boot entry as described in "Adding Kernel Arguments at Boot Time" in *Booting and Shutting Down Oracle Solaris 11.4 Systems*, and remove the property.

When the client is booted, a menu displays, as shown in "Starting an Automated Installation from the Command Line" on page 108. Use this menu to examine or install the system.

Starting an Automated Installation from the Command Line

If you selected a boot option that does not initiate an installation, then the following menu is displayed:

```
1  Install Oracle Solaris
2  Install Additional Drivers
3  Shell
4  Terminal type (currently xterm)
5  Reboot

Please enter a number [1]:
```

Select option 3 to open a shell.

Once a shell is running, use the following commands to start an automated installation:

```
$ svcadm enable svc:/application/manifest-locator:default
$ svcadm enable svc:/application/auto-installer:default
```

Preventing an Automatic Reboot

For debugging purposes, you might want to inspect the system after installation has been completed but before it is rebooted. You can override requests for automatic booting contained in a manifest file or in boot command arguments by creating the file `/system/volatile/noautoreboot`.

To prevent an automatic reboot after the installation, log in to the target client system and type the following command:

```
$ touch /system/volatile/noautoreboot
```

Then begin the automatic installation. At the end of the installation, the process checks for the presence of the file. If the file is detected, installation ends and you can begin diagnosing the system.

Note - Ensure that you have the proper privileges to create the file in the client system.

Index

B

C

D

E

T

U

V

W

X

Z