HP Server Automation

for the HP-UX, IBM AIX, Red Hat Enterprise Linux, Solaris, SUSE Linux Enterprise Server, VMware, and Windows® operating systems

Software Version: 7.80

Planning and Installation Guide

Document Release Date: June 2009
Software Release Date: June 2009

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Intel® Itanium® is a trademarks of Intel Corporation in the U.S. and other countries.

Java™ is a US trademark of Sun Microsystems, Inc.

Microsoft®, Windows®, Windows® XP are U.S. registered trademarks of Microsoft Corporation.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

1 SA Architecture

This section provides an overview of SA architecture. You will learn about the SA Core and its Core Components and the relationship between the core, Server Agents, and Satellites.

There is also a discussion of SA topologies which will help you decide on the topology for your SA installation.

Architecture Overview

SA provides a fully automated IT environment. IT teams are able to work together seamlessly, even if they are in different geographies. No matter what their location, all administrators have the same view of the IT environment.

At the simplest level, an SA installation consists of:

- The SA Core and its Core Components installed on a host server or servers
- A set of SA Gateways (Core, Agent, Management, Satellite) that enable communications between the SA Core and the Managed Servers
- SA Server Agents installed on Managed Servers

Each server in a Facility that is to be managed using SA must have a Server Agent installed. A *Facility* is a construct that typically represents a collection of servers that a single SA Core manages. The Core and its Core Components are installed on their own server (optionally, across multiple servers) and communicate through SA Gateways with the Agents on Managed Servers to provide centralized monitoring, reporting, and management capabilities.

New Architecture: SA Core Component Bundling

SA 7.50 introduced the concept of *Core Component bundling*.

SA 7.80 expands on bundled architecture with the move of the Software repository from the Infrastructure bundle to the Slice Component bundle and the introduction of the Software Repository Store.

In a typical installation, certain Core Components are *bundled* or grouped together and must be installed together on the same host. This architecture facilitates ease of installation and maintenance, adds simplicity and robustness for multi-server deployments, supports horizontal scaling and Core Component load balancing. For detailed information about Core Component bundling, see SA Core Component Bundling on page 15.

The SA Core

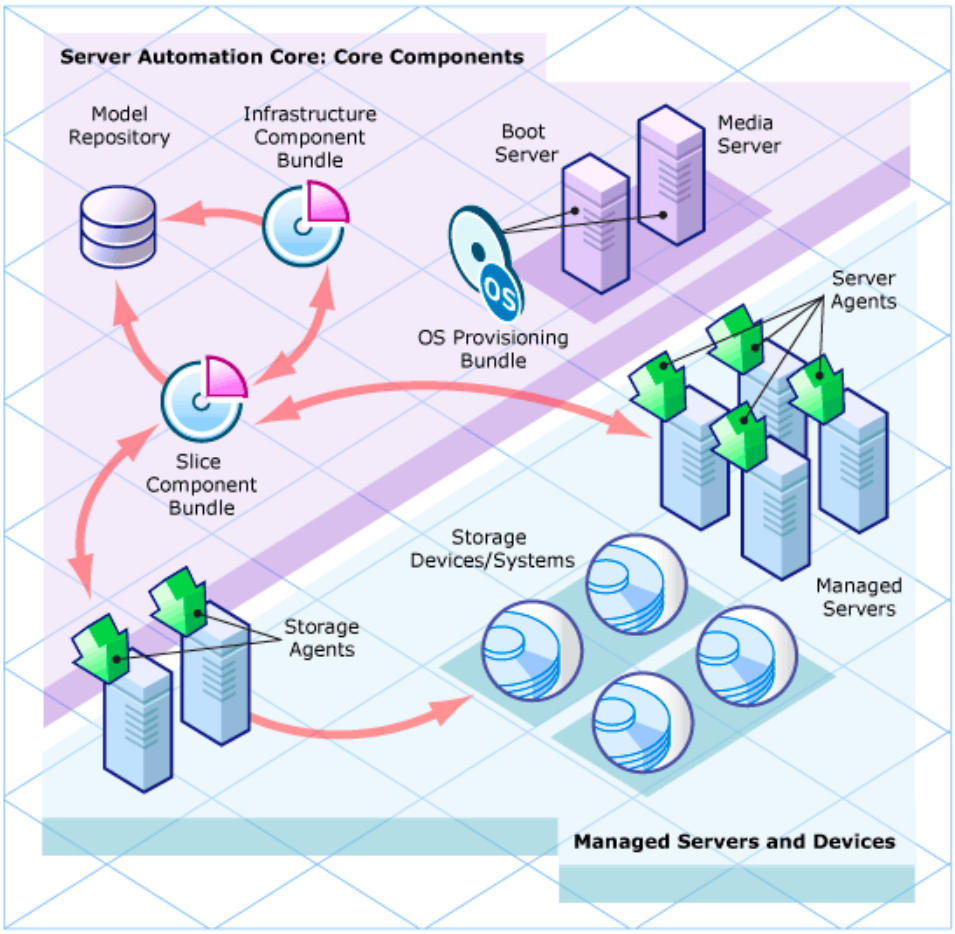
The *Core* is actually a set of *Core Components* that work together to allow you to discover servers on your network, add those servers to a Managed Server Pool, and then provision, monitor, configure, audit, and maintain those servers from a centralized SAS Web Client or SA Client. These Components provide management, communication, and OS provisioning capabilities, among other services.

The machines that the Core Components are installed on are called *Core Servers* or hosts. *Server Agents* are installed and reside on the Managed Servers and communicate with the Core and the Managed Servers through Gateways, and actually perform certain actions on the Managed Servers as directed by user input from the SA Client or SAS Web Client. These clients provide a GUI interface to the information and management capabilities of SA.

A Simple Single Core Installation

Figure 1 shows a simplified representation of a single core with all Managed Servers in the same facility, typically the First Core of a Multimaster Mesh. Most installations consist of multiple cores in different facilities. See SA Topologies on page 22.

Figure 1 An *SA* Core and Agents



A *Core Server* hosts the SA Core Components that allow SA to discover and store information about the location and configuration of all the servers on your network as well as components that perform monitoring, auditing, provisioning and maintenance tasks.

Certain Core components can be installed in the same instance across multiple servers while still being seen as a single logical entity.

SA Server Agents

An SA Server Agent is intelligent software that is installed on a server that you want to manage using SA. After an agent is installed on an unmanaged server, it registers with the SA Core which can then add that server to its pool of Managed Servers. The Server Agent also receives commands from the Core and initiates the appropriate action on its local server, such as software installation and removal, software and hardware configuration, server status reporting, auditing, and so on.

You can install agents on servers in the following ways:

- You can use the SA Deployment and Discover (ODAD) utility to discover the servers on your network that do not have SA Server Agents installed and then deploy the agents to those servers.
- You can use the SA OS Provisioning feature to provision an operating system to a bare-bones server — an SA Server Agent will also be installed.
- You can copy the SA Server Agent binary to the server and install it manually.

During agent registration, SA assigns each server a unique ID (the Machine ID (MID)) and stores this ID in the Model Repository. Servers can also be uniquely identified by their MAC Address (the network interface card's unique hexadecimal hardware identifier, which is used as the device's physical address on the network).

The Core Components

The Core Components are the heart of the SA Core making it possible to communicate with, monitor, and manage servers. Users and developers interact with the core through the SA Client or SAS Web Client, the command line, the API, and so on. Users can retrieve vital information about their network servers, provision servers, apply patches, take servers on and off line, configure and audit servers, and more. This interaction is controlled by the Core Components.

For example, a user could use the OS provisioning feature of the SA Client to identify an unprovisioned server, assign an OS Sequence to that server, and remotely begin the provisioning process.

The following section describes the SA Core Components and interfaces. For detailed information about how the SA Components work together to manage your servers, see the *SA Administration Guide*.

SA Core Component Bundling

The release of SA 7.80 expands on the concept of *SA Core Component Bundling* as a way of distributing Core Components in an SA installation introduced in SAS 7.0 and SA 7.50. Certain components are *bundled* together and must be installed as a *unit* during a Typical Installation. During a Custom installation, certain components can be broken out of their bundles (such as the Command Engine, the OS Provisioning Boot

Server and Media Server, among others) and installed on separate servers. For more information about Typical vs. Custom installations, see Chapter 6, “Installing the First Core” and Chapter 8, “Multimaster Mesh Installation”.

Component Bundling provides the following benefits:

- Added simplicity and robustness for multi-server deployments
- Scaling capability: you can install additional Slice” Components bundles for horizontal scaling
- Improved High Availability
- Load balancing between slices when multiple instances installed

New in SA 7.50:

- The SA Command Engine is installed as part of the Slice Component bundle, therefore you can have multiple Command Engine per core thus increasing SA 7.50’s ability to manage large numbers of servers simultaneously.

New in SA 7.80:

- The *Software Repository* is now installed as part of the *Slice Component bundle*, therefore you can have multiple Software Repositories per Core. Also new is the *Software Repository Store* which is part of the *Infrastructure Component bundle* and handles NFS exports to Slice Component bundle hosts.

Table 1 shows how components are bundled.

Table 1 Component Distribution

Model Repository	Infrastructure Components	OS Provisioning Components	Slice Components#1	Slice Components#2
One per core	One per core	Typically one per core	Multiple per core	Multiple per core
Model Repository	Management Gateway, Primary Data Access Engine Model Repository Multimaster Component Software Repository Store (can be located on another host)	Media Server Boot Server	Core Gateway/ Agent Gateway Command Center Global File System Web Services Data Access Engine Secondary Data Access Engine Build Manager Command Engine Software Repository	Core Gateway/Agent Gateway Command Center Global File System Web Services Data Access Engine Secondary Data Access Engine Build Manager Command Engine Software Repository

The *Boot Agent* is unrelated to Server Agents and operates as part of OS Provisioning.

Model Repository

The Model Repository is implemented as an Oracle database. It is a standalone component and is not bundled with other Core Components. All SA components work from or update a data model maintained for all servers that SA manages. The Model Repository contains essential information necessary to build, operate, and maintain the following items:

- An inventory of all servers under SA management.
- An inventory of the hardware associated with these servers, including memory, CPUs, storage capacity, and so on.
- Information about the configuration of the servers, including IP addresses.
- An inventory of the operating systems, system software, and applications installed on servers.
- An inventory of operating systems and other software that is available to be provisioned to the servers along with software policies that control how the software is configured and installed.
- Authentication and security information.

The Core Component Bundles

Infrastructure Component Bundle

- **Primary Data Access Engine**

  The Data Access Engine provides an XML-RPC interface to the Model Repository that simplifies interaction with various clients, such as the SAS Web Client, system data collection, and monitoring agents on servers. The Data Access Engine installed with the Infrastructure Component bundle is designated the *Primary* Data Access Engine. The Data Access Engine installed with the Slice Component bundle(s) is designated the *Secondary* Data Access Engine.

  Because interactions with the Model Repository go through the Data Access Engine, clients are less impacted by changes to the Model Repository's schema. The Data Access Engine allows features to be added to SA without requiring system-wide changes.

- **Management Gateway**

  Manages communication between SA Cores and between SA Cores and Satellites.

- **Model Repository Multimaster Component**

  The Model Repository Multimaster Component is installed with the Infrastructure Component bundle. A Multimaster Mesh, by definition, has multiple core installations and the Model Repository Multimaster Component synchronizes the data in the Model Repositories for all cores in the Mesh, propagating changes made in one repository to the other repositories.

  Each Model Repository Multimaster Component consists of a Sender and a Receiver. The Sender (Outbound Model Repository Multimaster Component) polls the Model Repository and sends unpublished transactions to other Model Repositories. The Receiver (Inbound Model Repository Multimaster Component) accepts the transactions from other Model Repositories and applies them to the local Model Repository.

As of SA 7.80, TIBCO Rendezvous has been replaced by the SA Bus. The SA Bus is a set of libraries that provide a certified messaging services.

- **Software Repository Store** (*Optional*)

  The Software Repository Store component can be installed on any server hosting an Infrastructure Component bundle. As of SA 7.80, the Software Repository is part of the Slice Component bundle and the Software Repository Store component has been introduced to handle NFS exports to Slice Component bundle hosts.

  If you choose not to install the Software Repository Store, you can manually configure an *optional* NAS (filer) to allow Slice Component bundle servers access to the filesystem.

Slice Component Bundle

- **Command Engine**

  The Command Engine is a system for running distributed programs across many servers (typically through SA Server Agents). Command Engine scripts are written in Python and run on the Command Engine server. Command Engine scripts can issue commands to Server Agents. These calls are delivered in a secure manner and are auditable by using data stored in the Model Repository.

  SA features (such as Code Deployment & Rollback) can use Command Engine scripts to implement part of their functionality.

  As of SA 7.50, the Command Engine was moved to the Slice Component bundle. Because you can have multiple Slice Component bundles, and therefore multiple Command Engines, horizontal scaling is greatly enhanced. Multiple Command Engine instances can share the load of command delivery and script execution by taking advantage of the load balancing mechanism provided by multiple Slice Component bundles. Failover and high availability are also improved. For example, when a Command Engine instance tries to delegate a command to another node in the cluster and that node is down, it fails over to the next node.

- **Software Repository**

  As of SA 7.80, the Software Repository was moved from the Infrastructure Component bundle to the Slice Component bundle. This component is a repository in which the binaries/packages/source for software/application provisioning and remediation is uploaded and stored. The Software Repository Store on the Infrastructure Component bundle handles NFS exports to Slice Component bundle hosts.

  For information about how to upload software packages to the Software Repository, see the *SA Policy Setter Guide*.

- **Core Gateway/Agent Gateway**

  The Core Gateway communicates directly withe Agent Gateways passing requests and responses to and from Core Components. Agent Gateways are installed on Managed Servers and communicate with the Core Gateway

- **Command Center**

  The Command Center (OCC) is the Core Component that underlies the SAS Web Client. The OCC includes an HTTPS proxy server and an application server. You access the OCC only through the SAS Web Client.

- **Global File System**

  The Global File System (OGFS) is installed with each Slice Component Bundle and provides the central execution environment for SA.

  The OGFS runs on one or more physical servers; customers can scale SA execution capacity by simply adding additional Slice Component bundles in a core.

  The OGFS runs SA built-in components — as well as customer-written programs — within a virtual file system that presents the SA data model, SA actions, and managed servers as virtual files and directories.

This unique feature of SA allows users of the Global Shell and Automation Platform Extensions (APX) to query SA data and manage servers from any scripting or programming language. Since the OGFS filters all data, actions, and managed server access through the SA security model, programs running in the OGFS are secure by default.

- **Web Services Data Access Engine**

  The Web Services Data Access Engine provides a public-object abstraction layer to the Model Repository and provides increased performance to other SA Core Components. This object abstraction can be accessed through a Simple Object Access Protocol (SOAP) API, through third-party integration components, or by a binary protocol of SA components such as the SAS Web Client.

- **Secondary Data Access Engine**

  The Data Access Engine provides an XML-RPC interface to the Model Repository that simplifies interaction with various clients, such as the SAS Web Client, system data collection, and monitoring agents on servers. The Data Access Engine installed with the Infrastructure Component bundle is designated the *Primary* Data Access Engine. The Data Access Engine installed with the Slice Component bundle(s) is designated the *Secondary* Data Access Engine.

  Because interactions with the Model Repository go through the Data Access Engine, clients are less impacted by changes to the Model Repository's schema. The Data Access Engine allows features to be added to SA without requiring system-wide changes.

- **Build Manager**

  Although the Build Manager is part of the OS Provisioning feature it is installed as part of the Slice Component bundle. The Build Manager facilitates communications between OS Build Agents and the Command Engine. It accepts OS Provisioning commands from the Command Engine. It provides a runtime environment for the platform-specific build scripts to perform the OS Provisioning procedures.

OS Provisioning Components Bundle

- **Boot Server**

  The Boot Server is part of the OS Provisioning feature. It supports network booting of Sun and x86 systems with inetboot and PXE, respectively. The processes used to provide this support include the Internet Software Consortium DHCP server, Sun Solaris TFTP, and NFS.

- **Media Server**

  The Media Server is part of the OS Provisioning feature. It is responsible for providing network access to the vendor-supplied media used during OS Provisioning. The processes used to provide this support include the Samba SMB server and Sun Solaris/Linux NFS. You copy and upload your valid operating system installation media to the Media Server.

OS Build Agent: The OS Build Agent is part of the OS Provisioning feature. It runs during the pre-provisioning (network boot) process and is responsible for registering a bare metal server with the SA Core through the Build Manager and guiding the OS installation process.

Satellite Installations

- **Software Repository Cache**

  A Software Repository Cache contains local copies of the contents of a Core's Software Repository (or of another Satellite). Having a local copy of the Software Repository can improve performance and decrease network traffic when you install or update software on a Satellite's Managed Servers.

- **Satellite Agent**

  The Satellite Agent handles communications between the Satellite and the Core through the Core's Management Gateway.

SA Interfaces

SAS Web Client

The SAS Web Client is an HTML browser-based user interface to SA through which users can:

- Manage servers
- Deploy code and content to servers (deprecated)

SA Client

A Java™ Web-Start cross-platform application that extends the SAS Web Client features and provides the following features:

- **Configure Software Policies**
- **Provision software/applications/packages onto Managed Servers**
- **Provision operating systems onto bare metal servers**
- **Run distributed scripts on servers**
- **Discovery and Agent Deployment**
- **Device Explorer**, to provide detailed hardware information
- **Virtualization Director**, to manage your virtualized installations
- **Server Automation Visualizer** (SAV), to manage the operational architecture and behavior of your distributed business applications
- **Audit and Remediation**, to track compliance
- **Compliance Dashboard**
- **Reports**
- **Software Management**
- **Patch Management for Windows**
- **Patch Management for Unix**
- **Application Configuration Management**
- **Global Shell**
- **NA Integration**

SA Command Line Interface (OCLI)

A command line interface used to upload packages into the Software Repository, and to perform batch commands, run scripts, and many other SA operations.

DCML Exchange Tool (DET)

A utility that enables users to export almost all server management content from any SA Core and import it into any other SA Core.

ISM Development Kit

A development kit that consists of command-line tools and libraries for creating, building, and uploading ISMs. An ISM is a set of files and directories that include application bits, installation scripts, and control scripts.

SA APIs

A set of APIs and a command-line interface (CLI) that facilitate the integration and extension of SA. This platform allows other IT systems — such as existing monitoring, trouble ticketing, billing, and virtualization technology — to exchange information with SA. This broadens the scope of how IT can use SA to achieve operational goals.

For more information about all the interfaces, see the *SA Administration Guide*.

SA Gateways

SA Gateways manage communication between Managed Servers and a SA Core, between multiple cores, and between Satellite installations and a SA Core. Multimaster installations are discussed in Multimaster Mesh (Multiple Cores) on page 22 and Satellite installations are discussed in Multimaster Mesh (Multiples Cores and Satellites) on page 23.

There are several types of gateways:

- **Management Gateway**

  This gateway manages communication between SA Cores and between SA Cores and Satellites.

- **Core Gateway/Agent Gateway**

  These gateways work together to facilitate communication between the SA Core and Server Agents.

- **Satellite Gateway**

  This gateway communicates with the SA Core through the Management Gateway.

SA Topologies

You must decide what SA topology fits your facility's needs. This section provides some background on the SA topologies to help you make that decision

Single Core

The simplest topology is a Single Core (formerly a Standalone Core) that manages servers in a single facility.

A Single Core is best for a small network of servers contained in a single facility. Although a Single Core does not communicate with other SA Cores, it has all the components required to do so and can be easily converted into a core that is part of a Multimaster Mesh.

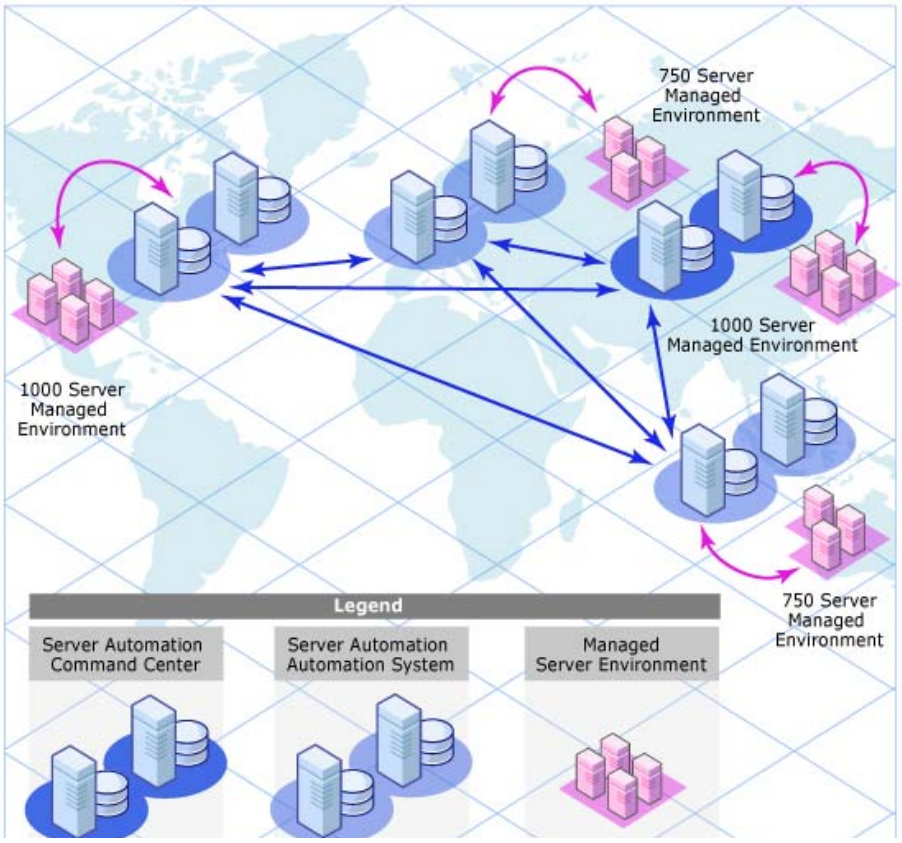
After the core is installed, you can use the Deployment and Discover (ODAD) utility to discover the servers on your network that do not have Server Agents installed and then deploy agents to those servers. After the Server Agents are deployed, they will automatically contact the Core through the Agent Gateway and register the server they are installed on with SA.

You can then use the SA Client to manage your servers.

Multimaster Mesh (Multiple Cores)

To manage servers in more than one facility, you should install a Multimaster Mesh of SA Cores or a combination of SA Cores and Satellites.

Figure 2 Multimaster Topology



A *Multimaster Mesh* is a set of two or more SA Cores that communicate through Management Gateways and can perform synchronization of the data about their Managed Servers contained in their respective Model Repositories over the network. Changes to the data in any Model Repository in a Multimaster Mesh are broadcast to all other Model repositories in the Mesh.

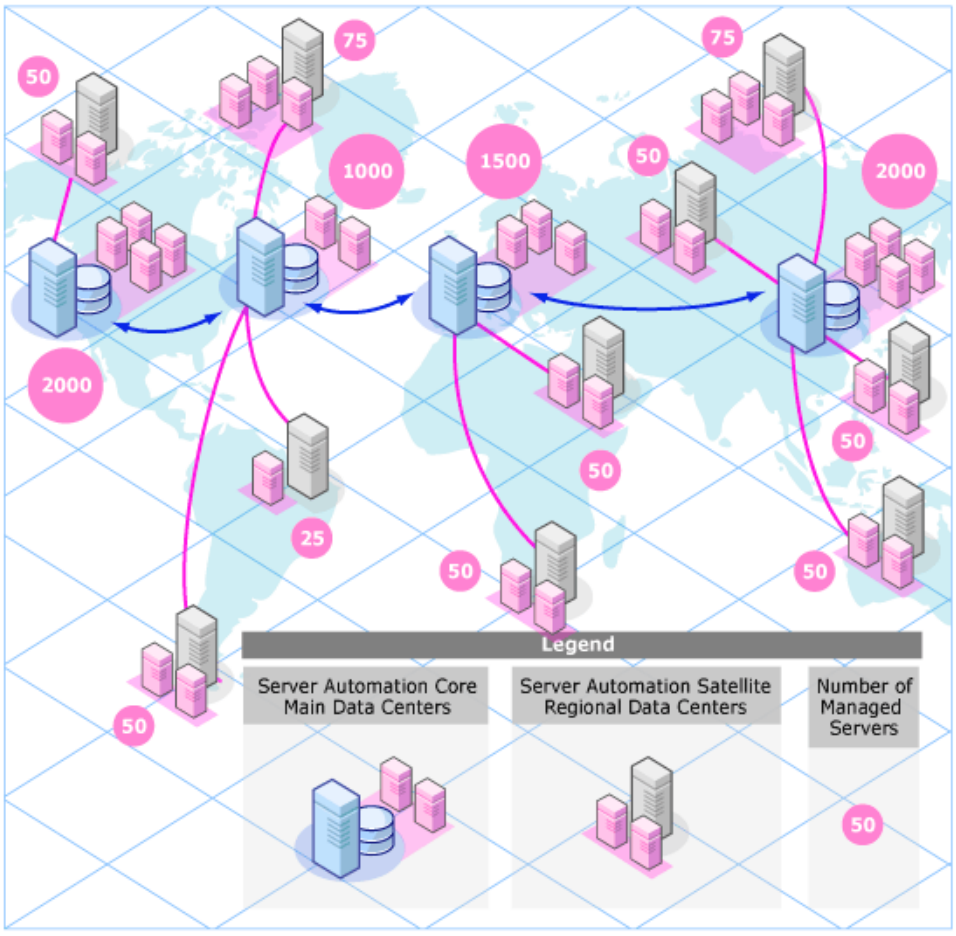
The SA Core Component that propagates and synchronizes changes from each model repository database to all other model repository databases is called the Model Repository Multimaster Component. This replication capability allows you to store and maintain a blueprint of software and environment characteristics for each facility making it easy to rebuild your infrastructure in the event of a disaster. It also provides the ability to easily provision additional capacity, distribute updates, and share software builds, templates and dependencies across multiple facilities — all from a single user interface.

Multimaster Mesh (Multiples Cores and Satellites)

A Multimaster Mesh can also include Satellite installations as shown in Figure 3.

Los Angeles, New York, London, and Tokyo have SA Core installations and each facility links to one or more Satellite installations in smaller facilities some in a star formation, others in cascading Satellite formation. See SA Satellites on page 26.

Figure 3 Server Management in Multiple Facilities with Satellites



Servers can be managed from any facility with an installed SA Core using the SAS Web Client or the SA Client. Using the example in Figure 3, a user can log on to the SA Client at the New York facility and manage servers that belong to the Los Angeles facility as long as he has the appropriate access rights and privileges.

Benefits of Multimaster Mesh

An Multimaster Mesh offers the following benefits among others:

- **Centralized Administration** — the Managed Servers in a Multimaster Mesh can be centrally administered from any facility with a Core installation. Administration is not locked into a single location or even restricted geographically.
- **Redundancy** — Synchronized (replicated) data management between facilities provides redundancy. For example, if the SA Core in one facility is damaged, another core in the Multimaster Mesh will contain a synchronized copy of the managed server data that can be used to restore the damaged core's Model Repository to a last known good state. In addition, while a damaged core is unavailable, other cores in the mesh can continue functioning without interruption.

  Replication also provides the ability to close down or add a facility while other facilities in the mesh continue operations without interruption.
- **Performance Scalability** — In a Multimaster Mesh, only multimaster database synchronizations are transmitted over the network reducing network bandwidth load.
- **Geographic Independence** — Cores can continue to manage servers during network interruptions regardless of location.

Facilities and Realms

SA Gateways use two constructs that facilitate routing network traffic and eliminate the possibility of IP address conflicts:

Facilities

A *Facility* is a construct that typically represents a collection of servers that a single SA core manages through the data about the managed environment stored in its Model Repository. A facility typically represents a specific geographical location, such as Sunnyvale, San Francisco, or New York, or, commonly, a specific data center.

A Facility is a permissions boundary within SA, that is, a user's permissions in one Facility do not carry over to another. Every Managed Server is assigned to a single facility. When a device initially registers with the SA Core, it is assigned to the facility associated with the gateway through which it is registering.

For example, Admin A works in Sunnyvale and is in charge of maintaining server patches. In a Facility framework, Admin A is bound to the Sunnyvale Facility as a user. When Admin A views servers, only those servers that are also bound to the Sunnyvale Facility are displayed. He will not see servers for any other Facility.

There are two types of facilities

- **Core Facilities**

  There is one Core Facility for every SA Core installation.
- **Satellite Facilities**

  A default Facility created when you install a Satellite.

Realms

Realms are a SA concept that allow SA to manage servers on different networks in the same Facility without fear of IP address conflicts.

A Realm is a logical entity that defines an IP namespace *within which* all Managed Server IP addresses must be unique. However, servers that are assigned to a *different Realms* can have duplicate IP addresses and still be uniquely identified within SA by their Realm membership.

Realms are interconnected by gateways in what can be described as a *gateway mesh* — a single interconnected network of SA Gateways.

When you create and name a new Facility during installation, a *default* Realm is also created with the same name as the Facility. For example, when you create the Facility, *Datacenter*, the installation also creates a Realm named *Datacenter*. Subsequent Realms in that facility could be named *Datacenter001*, *Datacenter002*, and so on. IP address in each realm are uniquely identified by the combination of the Realm name and the IP address, eliminating any problem with duplicate IP addresses in the same Facility.

Multimaster Mesh Topology Examples

Figure 4 shows a Multimaster Mesh with cores installed in two separate facilities, San Francisco and Los Angeles. Each facility's core has a Model Repository that contains data about the Managed Servers in both facilities. That data is constantly synchronized (replicated) between both Facilities' Model Repositories. The cores communicate through their respective Management Gateways.

Communication from the Managed Servers in the Los Angeles facility to the San Francisco core travels through the Los Angeles Agent Gateway to the Core Gateway, then to the Los Angeles Management Gateway which then communicates with the San Francisco core through the San Francisco Management Gateway and Core Gateway.

Figure 4 Multimaster Mesh with Two Cores

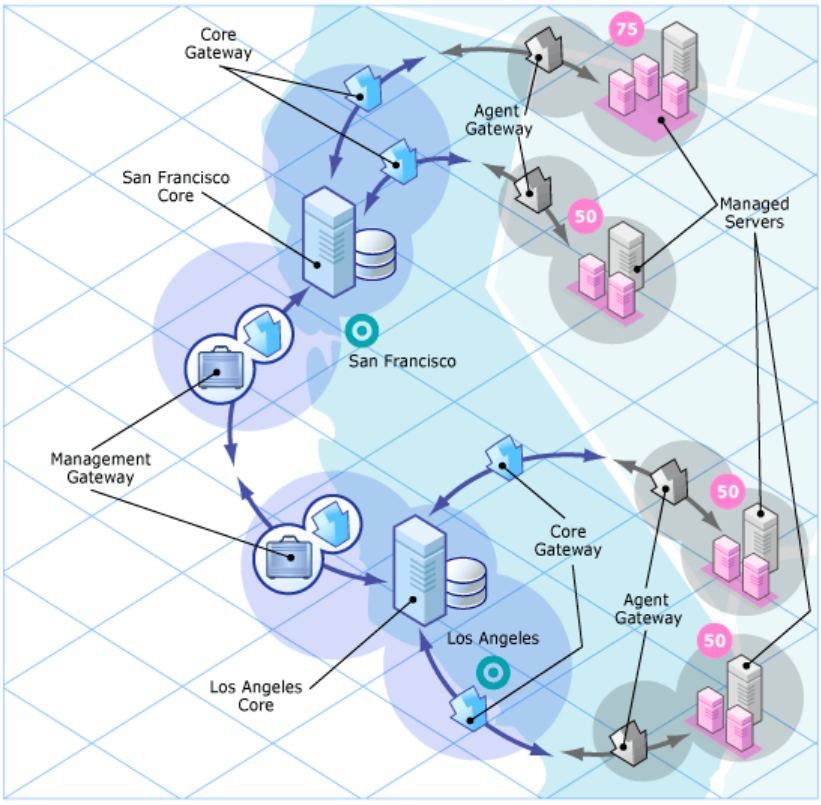
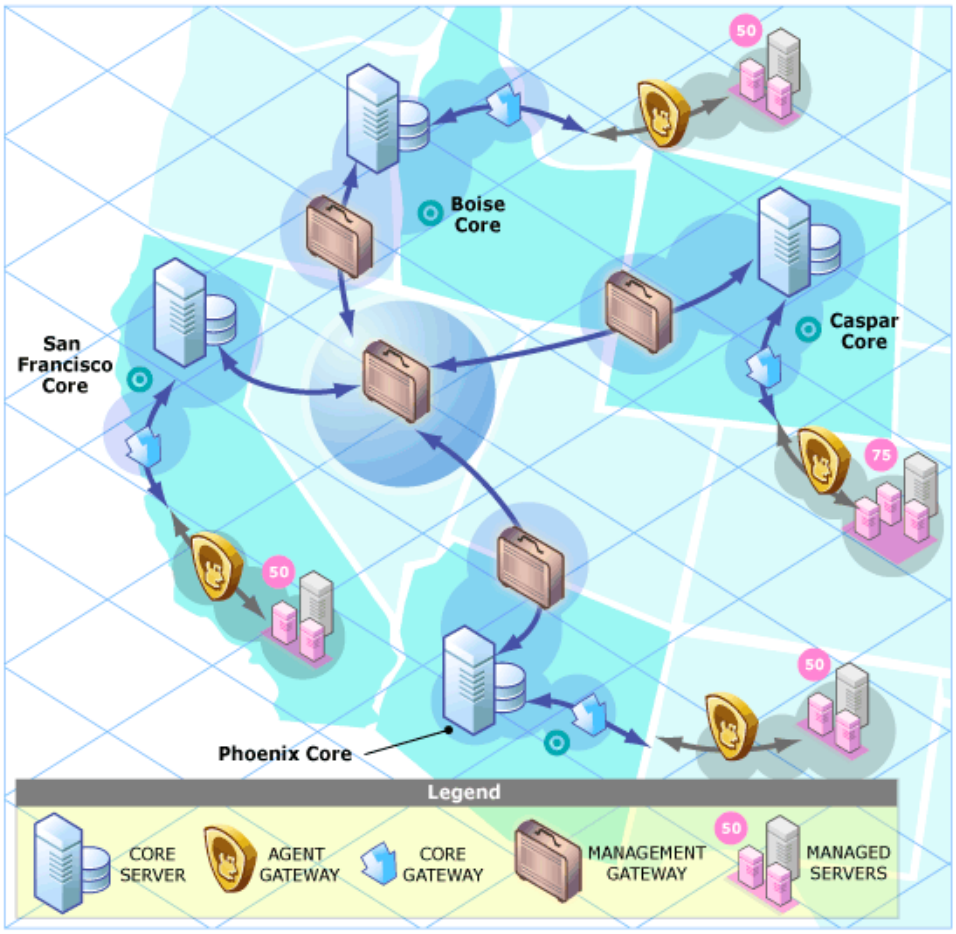


Figure 5 shows a Multimaster Mesh with four cores. This Mesh topology is called a *Star Formation* with the San Francisco core at the center of the Mesh. The HP BSA Installer configures a Multimaster Mesh with a star topology by default.

Figure 5 Multimaster Mesh with Four Cores



SA Satellites

A Satellite installation can be a solution for remote sites that do not have a large enough number of potentially Managed Servers to justify a full SA Core installation. A Satellite installation allows you to install only the minimum necessary Core Components on the Satellite host which then accesses the Primary Core's database and other services through an SA Gateway connection.

A Satellite installation can also relieve bandwidth problems for remote sites that may be connected to a primary facility through a limited network connection. You can cap a Satellite's use of network bandwidth to a specified bit rate limit. This allows you to insure that Satellite network traffic will not interfere with your other critical systems network bandwidth requirements on the same pipe.

A Satellite installation typically consists of, at minimum, an Satellite Gateway and a Software Repository Cache and still allows you to fully manage servers at a remote facility. The Software Repository Cache contains local copies of software packages to be installed on Managed Servers in the Satellite while the Satellite Gateway handles communication with the Primary Core.

You can optionally install the OS Provisioning Boot Server and Media Server on the Satellite host to support remote OS Provisioning. Installing other components on the Satellite host is not supported.

For more information about Satellite installations, see Chapter 9.

Satellite Topology Examples

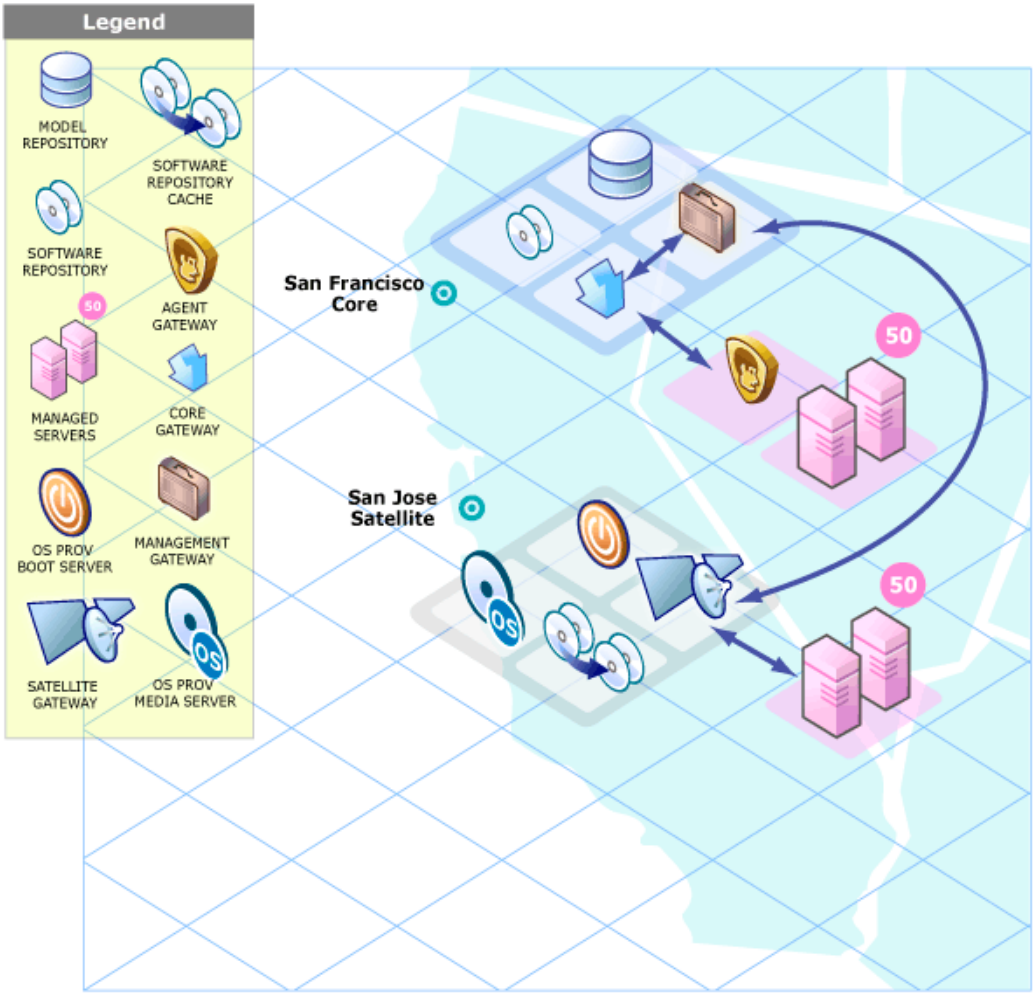
A Simple Single Core to Satellite Link

Figure 6 shows a single Satellite linked to a Single Core. In this example, the main facility is in San Francisco, and a smaller remote facility is in San Jose.

The San Francisco Single Core consists of several components, including the Software Repository, the Model Repository, an Agent gateway and a Management Gateway. For simplicity, this figure does not show all required Core Components, such as the Command Engine.

The San Jose Satellite consists of a Software Repository Cache, an Satellite Gateway, and an optional OS Provisioning Boot server and Media Server.

Figure 6 Satellite with the Single Core



For a more detailed description of these SA components, see Software Repository Cache on page 19, Boot Server on page 19, and Media Server on page 19.

The San Jose Satellite's Software Repository Cache contains local copies of software packages to be installed on Managed Servers in that facility.

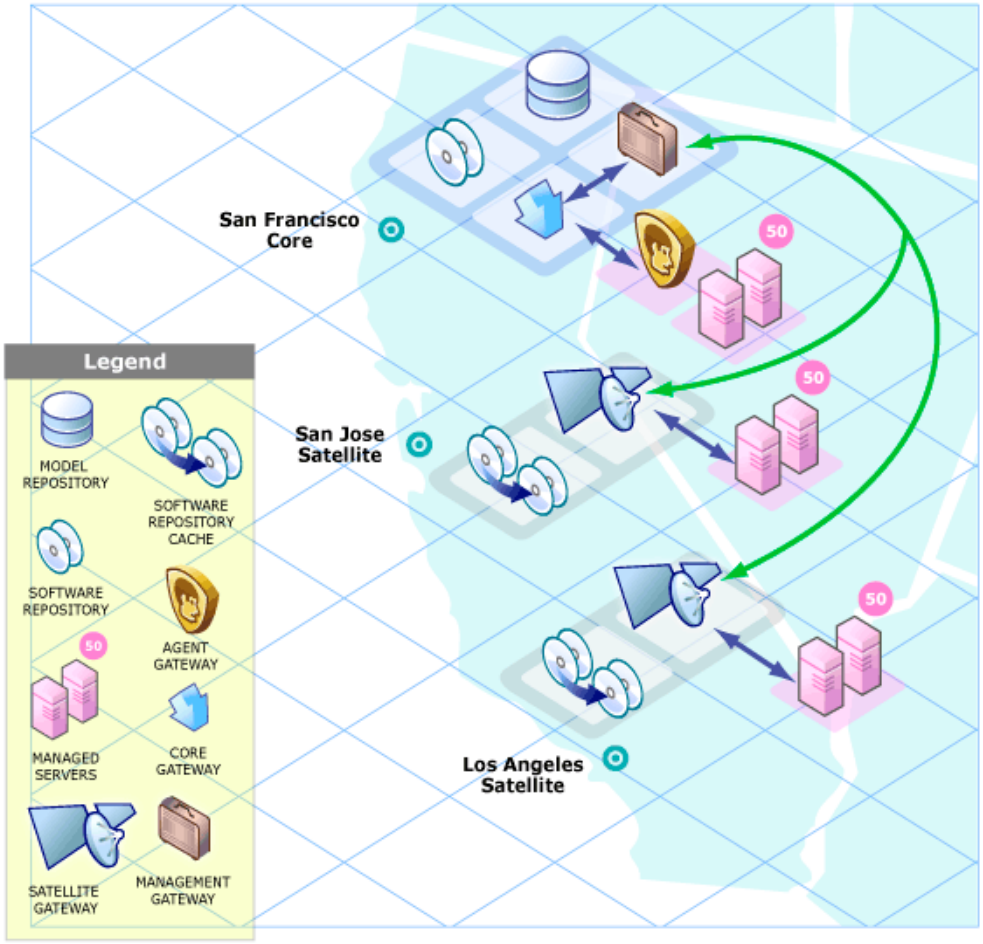
The Server Agents installed on managed servers at the San Jose facility connect to the San Francisco core through the San Jose Satellite Gateway which communicates with the San Francisco Management Gateway, then through the San Francisco Core gateway, ultimately, with the required Core Components.

Return communication reverses that path. The Server Agents installed on managed servers in the San Francisco facility communicate with the Core Components through the San Francisco facility's Agent and Core Gateways.

A Two Satellite to Single Core Link

Figure 7 shows two Satellites linked to a Single Core. In this example, San Francisco is the main facility, Sunnyvale and San Jose are Satellite facilities.

Figure 7 Two Satellites with a Single Core

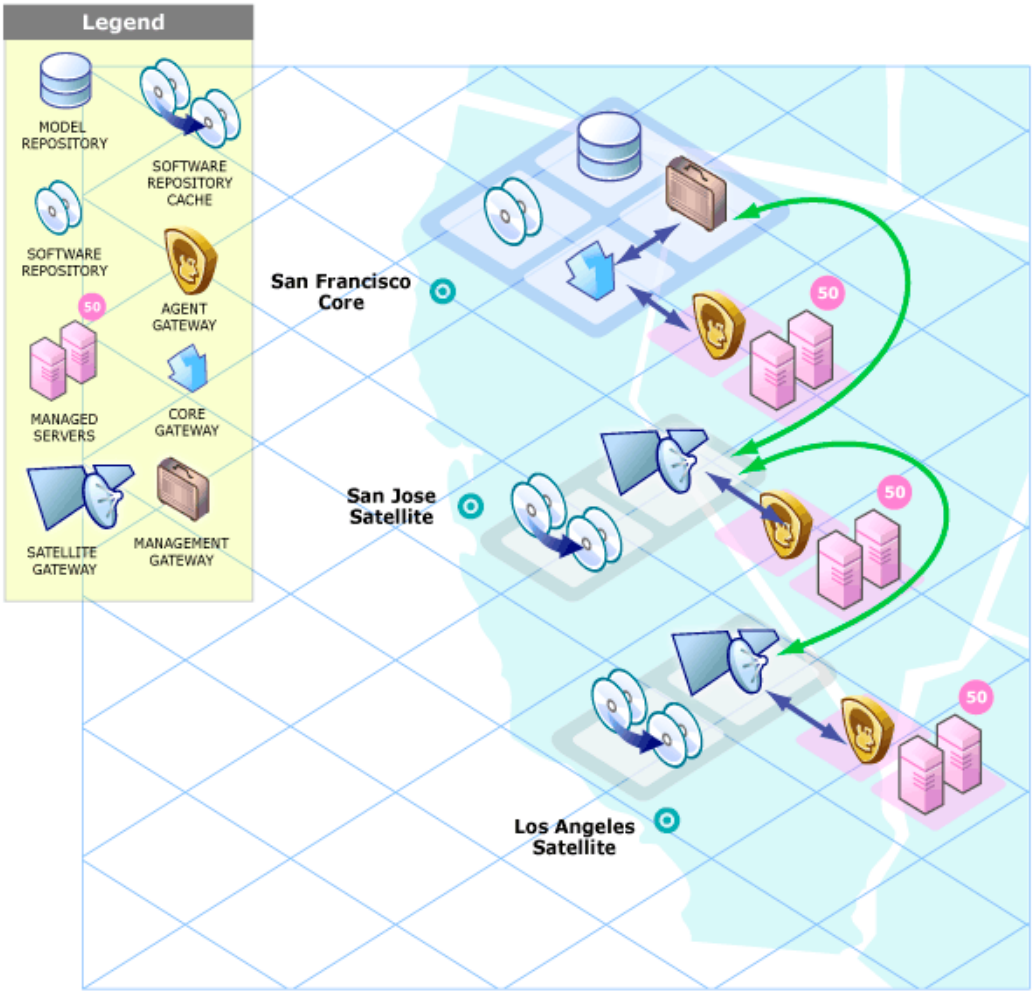


A Cascading Satellite Link

Figure 8 shows cascading Satellites, a topology in which Satellite Gateways are connected in a *chain*. This topology enables you to create a hierarchy of Software Repository Caches. Note that, the Satellite Gateways in this topology must belong to different SA Realms.

When tasked to install a package on a managed server in the Sunnyvale facility, SA first checks to see if the package resides in the Software Repository Cache in Sunnyvale. If the package is not in Sunnyvale, then SA checks the Software Repository Cache in San Jose. Finally, if the package is not in San Jose, SA goes to the Software Repository in the San Francisco core. For more information, see “Managing the Software Repository Cache” in the *SA Administration Guide*.

Figure 8 Cascading Satellites with a Single Core



Satellites in a Multimaster Mesh

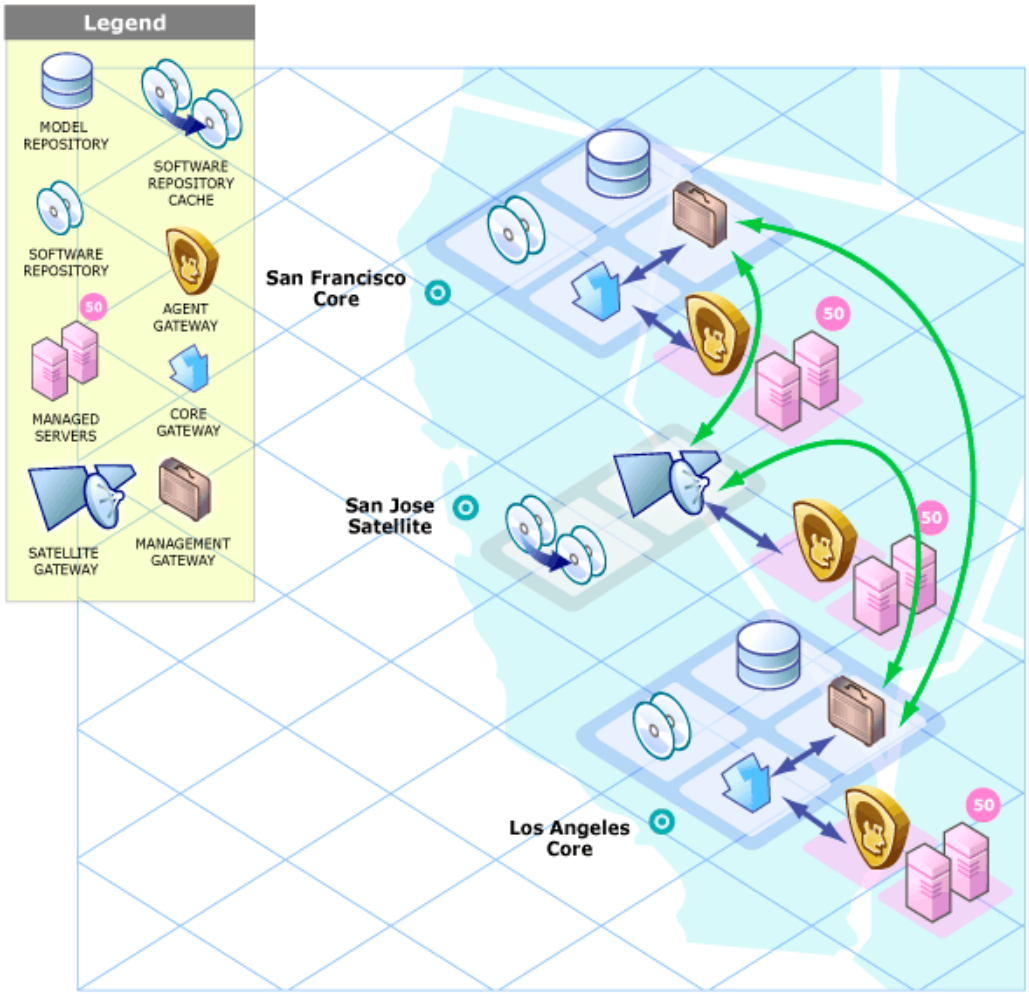
Figure 9 shows the San Jose Satellite connected to two SA Cores in a Multimaster Mesh.

Even when communication is possible to both Los Angeles and San Francisco, the Management Gateway chooses the route with the lowest cost (in Figure 9, the San Francisco route). You control cost evaluation using a parameter specified during Gateway installation. System designers can specify rules governing which SA Gateway routes to use to minimize network connectivity costs.

Using the same example environment in a failover scenario, during normal operations, the servers in the San Jose Satellite are managed by the San Francisco Core. Note, however, that the San Francisco and the Los Angeles Cores are directly connected through their Management Gateways.

If the connection between the San Jose Satellite and the San Francisco Core fails, the San Jose Satellite Gateway can immediately move communications from San Francisco to the Los Angeles core, allowing that core to maintain management of the San Jose servers. The Los Angeles Core will have up-to-date information about the San Jose site because the San Francisco Core's Model Repository data will have been replicated to the Los Angeles Model Repository as a part of normal SA operations.

Figure 9 Satellite in a Multimaster Mesh



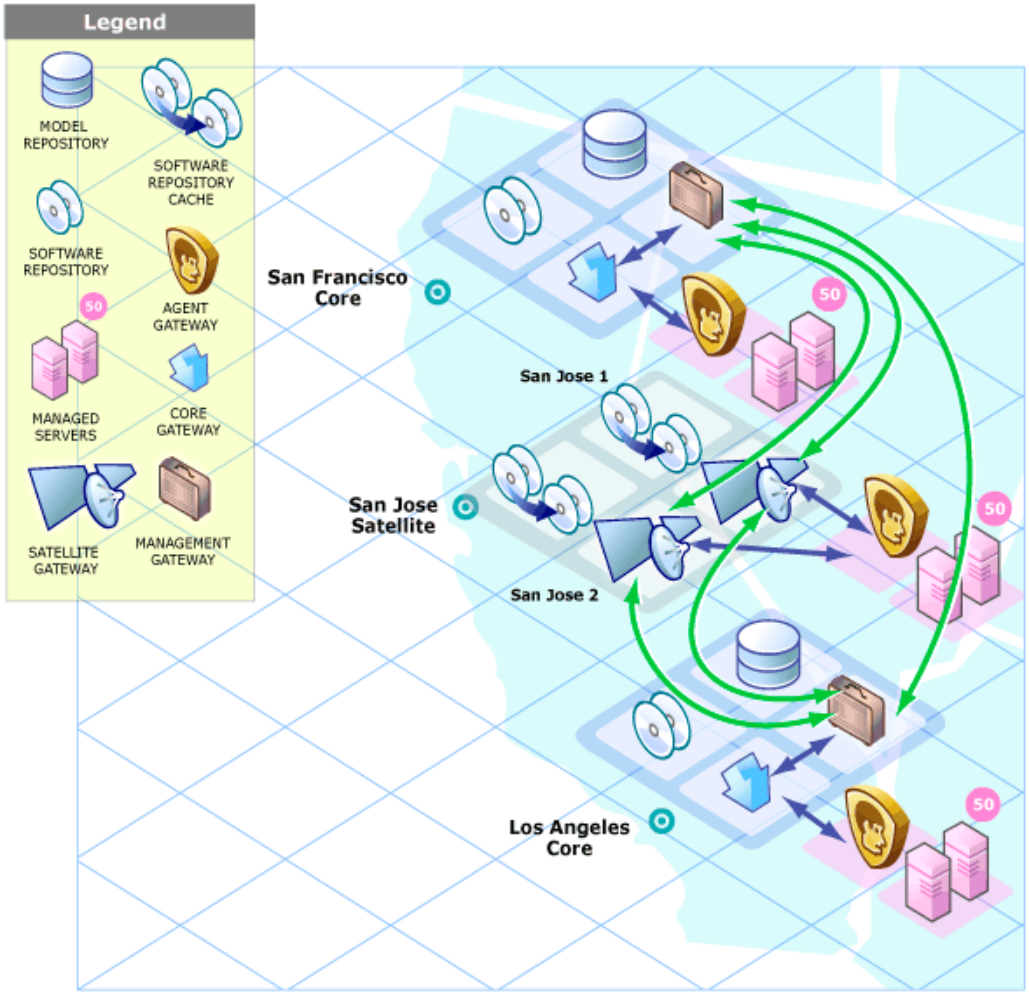
Satellite With Multiple Gateways in a Multimaster Mesh

Figure 10 shows a topology that provides failover capability in two ways. First, the San Jose Satellites 1 and 2 have Gateway connections to both the San Francisco and Los Angeles Management Gateways. If the Los Angeles core becomes unavailable, the San Francisco core can still manage the servers in the San Jose Satellite.

Second, the Agents installed on the Managed Servers in the San Jose Facility point to both of the Satellite's Agent Gateways. SA Agents automatically load balance over the available Agent Gateways and therefore can communicate directly with either the San Francisco or Los Angeles cores.

If one Gateway becomes unavailable, the Agents that are using the unavailable gateway as their primary gateway will automatically failover to using the secondary gateway. During routine agent-to-core communication, SA Agents will discover new gateways added to (or removed from) the Satellite.

Figure 10 Satellite With Multiple Gateways in a Multimaster Mesh



2 Operating System and Hardware Requirements

This section describes the supported operating systems for SA Core Servers, Managed Servers, and the SAS Java™ Client. This chapter also describes the hardware requirements for the servers running an SA Core and provides guidelines on how to distributeSA Core Components across one or more servers.

Supported Operating Systems: SA Server Agents and the SA Client

This section lists the supported operating systems for Server Agents and the SA Client.

SA Server Agent Supported Operating Systems

The following table lists the supported operating systems for SA Server Agents, which run on the servers managed by SA. Platforms with a (*) are deprecated in this release. For more information on platform deperecation in SA in this releasse, see Operating System Deprecation and End of Support on page 27.

Table 2 SA Server Agent Supported Operating Systems

Supported Operating Systems for SA Server Agent	Versions	Architecture
AIX		
	AIX 4.3*	POWER
	AIX 5.1*	POWER
	AIX 5.2	POWER
	AIX 5.3	POWER
	AIX 6.1	POWER
HP-UX		
	HP-UX 10.20*	PA-RISC (direct, nPartition)
	HP-UX 11.00*	PA-RISC (direct, nPartition)
	HP-UX 11.11	PA-RISC (direct, nPartition)
	HP-UX 11.23 (11i v2)	PA-RISC (direct, nPartition) Itanium (direct, nPartition)
	HP-UX 11.31 (11i v3)	PA-RISC (direct, nPartition) Itanium (direct, nPartition)

Table 2 SA Server Agent Supported Operating Systems (cont'd)

Supported Operating Systems for SA Server Agent	Versions	Architecture
Sun Solaris		
Note: Solaris 6, 7, 8, 9, and 10 are supported running on Sun Dynamic System Domains.		
	Solaris 6*	Sun SPARC 4u
	Solaris 7*	Sun SPARC 4u
	Solaris 8*	Sun SPARC 4u
	Solaris 9*	Sun SPARC 4u
	Solaris 10 (Updates 1, 2, 3, 4, 5, and 6)	Sun SPARC (4u, 4v) and x86_64, x86_32 Solaris 10 is supported in guest logical domains.
Fujitsu Solaris		
	Solaris 8*	Fujitsu SPARC
	Solaris 9*	Fujitsu SPARC
Windows		
	Windows NT 4.0*	x86_32
	Windows 2000 Server Family	x86_32
	Windows Server 2003	x86_32, x86_64
	Windows XP Professional	x86_32
	Windows Server 2008	x86_32, x86_64
Red Hat Linux		
	Red Hat Enterprise Linux 2.1* AS	x86_32
	Red Hat Enterprise Linux 2.1* ES	x86_32
	Red Hat Enterprise Linux 2.1* WS	x86_32
	Red Hat Enterprise Linux 3 AS	x86_32, x86_64, Itanium
	Red Hat Enterprise Linux 3 ES	x86_32, x86_64, Itanium
	Red Hat Enterprise Linux 3 WS	x86_32, x86_64, Itanium
	Red Hat Enterprise Linux 4 AS	x86_32, x86_64, Itanium
	Red Hat Enterprise Linux 4 ES	x86_32, x86_64, Itanium
	Red Hat Enterprise Linux 4 WS	x86_32 and x86_64, Itanium
	Red Hat Enterprise Linux Server 5	x86_32, x86_64, Itanium
	Red Hat Enterprise Linux Desktop 5	x86_32 and x86_64

Table 2 SA Server Agent Supported Operating Systems (cont'd)

Supported Operating Systems for SA Server Agent	Versions	Architecture
SUSE Linux		
	SUSE Linux Enterprise Server 8*	x86_32
	SUSE Linux Standard Server 8*	x86_32
	SUSE Linux Enterprise Server 9	x86_32 and x86_64
	SUSE Linux Enterprise Server 10	x86_32 and x86_64
VMware		
	ESX Server 3.0	x86_32 and x86_64
	ESX Server 3.5	x86_32 and x86_64
	ESXi Server 3.5	x86_32 and x86_64

On Red Hat Enterprise Linux 4 AS, SA does not support SELinux (Security Enhanced Linux). By default, SELinux is enabled on Red Hat 4 AS. You must disable the SELinux feature on Red Hat 4 AS for the SA Agent to function correctly. SA supports SELinux (Security Enhanced Linux) on Enterprise Linux 5.

Required Operating System Packages/Patches for SA Agents

Table 3 Required Packages/Patches for SA Agents

Server operating system	Required patches
AIX 4.3 AIX 5.1	APAR IY39444 APAR IY39429 NOTE: If AIX 4.3.3.388, 4.4.4.89, or 5.1.0.3 is installed, the Agent Installer displays an error message that indicates the correct APAR to install on the server.
HP-UX (10.20, 11.00, 11.11/11i)	For HP-UX 10.20, PHCO_21018 Additionally, SW-DIST should be upgraded to the HP recommended patch level. You should continue to upgrade this package when HP recommends new versions.
Linux AS 3.0 Linux WS 3.0 Linux ES 3.0	Red Hat Enterprise Linux 3 Update 3

Table 3 Required Packages/Patches for SA Agents (cont'd)

Server operating system	Required patches
Solaris 10, 9, 8, 7, and 6	SUNWadmc SUNWcsl SUNWcslr (If available, depending on version) SUNWcsu SUNWesu SUNWlibms SUNlibmsr (If available, depending on version) SUNWswmt It is strongly recommended not to remove packages from the SUNWCreq minimal required install cluster, since many packages are interdependent and operation beyond that of basic SA functionality may be affected.
Windows 2000	Service Pack 4
Windows NT 4.0	Service Pack 6a

SA Client Supported Operating Systems

Table 4 lists the operating systems supported for the SA Client.

Table 4 SA Client Supported Operating Systems

Supported Operating Systems for the SA Client	Versions	Architecture
Windows		
	Windows Vista	x86_32 and x86_64
	Windows XP	x86_32
	Windows 2003	x86_32
	Windows 2000	x86_32

A minimum of 1GB RAM on the system that runs the SA Client is necessary for optimal performance.

Agent Installation on Windows Server 2000 and Windows Server 2003

Installation of an SA Agent on a managed server requires the Windows Update service to be installed.

- If the service is installed, but has been disabled by the customer, the Agent will automatically start the service.
- If the service is not installed, the Agent will copy the Windows Update Agent installer to the managed server and then run it. This process will install the service and set it to automatically start on all deployed servers.

For information about the installer files for Patch Management, see Windows Patch Management Requirements on page 62.

If the Windows Update service is prevented from running when the Agent triggers the service to start (such as, when the service is blocked by a domain policy), the following error will be reported in the managed server system log:

```
DCOM got error "The service cannot be started, either because it is
disabled or because it has no enabled devices associated with it. "
attempting to start the service wuauserv with arguments "" in order to run
the server:
{E60687F7-01A1-40AA-86AC-DB1CBF673334}
```

For more information about this error, see **http://go.microsoft.com/fwlink/events.asp**.

Supported Operating Systems: SA Core Server

Table 5 lists the supported operating systems for SA Core and Satellite Components.

For a list of supported Oracle versions for the Model Repository, see Appendix A in the *SA Planning and Installation Guide***.**

SA Core Supported Operating Systems

Platforms with a (*) are deprecated in this release. For more information on platform deperecation in SA in this releasse, see Operating System Deprecation and End of Support on page 27.

Table 5 SA Core Supported Operating Systems

Supported os for SA core	Versions	Architecture	SA Components
Sun Solaris			
Sun Solaris	Solaris 10 (Updates 1, 2, 3, 4, 5 and 6)	Sun SPARC 4u, 4v(*) **Note**: The SA Core is also supported on Sun Dynamic System Domains.	All components
Red Hat Linux			
	Red Hat Enterprise Linux 3* AS	x86_32	All components
	Red Hat Enterprise Linux 4 AS	x86_64	All components
	Red Hat Enterprise Linux Server 5.2, 5.3	x86_64	All components
SUSE Linux			
	SUSE Linux Enterprise Server 10 SP2	x86_64	All components

SA 7.8 cores running on VMware ESX VM's are supported when specific requirements are met. For details on these requirements please download and read the Technical Note: SA Cores on VMWare VMs located at http://h20230.www2.hp.com/selfsolve/manuals.

A guest OS (virtual machine) of a VMWare ESX server *is not supported* as an SA Core server.

SA Core servers may not be installed in Solaris Local Zones and Solaris Local Zones may not be installed on a server where an SA Core is installed.

SA Satellite Supported Operating Systems

Table 6 lists the supported operating systems for these SA Satellite Components:

- Satellite Gateway
- Software Repository Cache
- OS Provisioning Boot Server (*optional*)
- OS Provisioning Media Server (*optional*)

Platforms with a (*) are deprecated in this release.

Table 6 SA Satellite Supported Operating Systems

Supported operating systems for SA satellite	Versions	Architecture
Sun Solaris		
	Solaris 10 (Updates 1, 2, 3, 4, 5 and 6) **Note**: The SA satellite is supported on Sun Dynamic System Domains.	Sun SPARC 4u, 4v*
Red Hat Linux		
	Red Hat Enterprise Linux 3* AS	x86_32
	Red Hat Enterprise Linux 4 AS	x86_64
	Red Hat Enterprise Linux Server 5.2, 5.3	x86_64
SUSE Linux		
	SUSE Linux Enterprise Server 9*	x86_32
	SUSE Linux Enterprise Server 10 SP2	x86_64

Veritas File System (VxFS)

SA does not currently support the Veritas File System (VxFS). If you attempt to install SA components on a system running VxFS, the installation will fail and need to be backed out.

Disk Space Requirements

An SA *Core Server* is a computer hosting one or more *SA Core Components*. You have the option to install all of the SA Core Components on a single server or distribute them across multiple servers. This section describes the hardware requirements for any SA Core Server.

Core Server Disk Space Requirements

On each Core Server, the root directory must have at least 72 GB available hard disk space. SA components are installed in the `/opt/opsware` directory. Table 7 lists the recommended disk space requirements for installing and running SA Core Components. These sizes are recommended for the primary production data. Additional storage for backups must be calculated separately.

Table 7 SA Disk Space Requirements

SA Component Directory	Recommended Disk Space	Requirement Origin
/etc/opt/opsware	50 MB	Configuration information for all SA Core services. (Fixed disk usage)
/media*	15 GB	**OS Provisioning**: The media directory holds the OS installation media that is shared over NFS or CIFS. The initial size for this directory depends on the total size of all OS installation media sets that you plan on provisioning, such as Windows Server 2003 CD (700mb), Red Hat AS3 CDs (2GB), and SUSE 9 SP3 (10GB). The network OS install shares do not need to reside on SA core systems and are typically dispersed across multiple servers as the Multimaster Mesh grows. (Bounded disk usage that grows quickly in large increments)
/opt/opsware	15 GB	The base directory for all SA Core services. (Fixed disk usage)
/u01/oradata /u02/oradata /u*nn*/oradata ...	20 GB	The Oracle tablespace directory that contains all model and job history information. Known sizes range from 5GB to 50GB of space, depending on the frequency and type of work, the amount of software and servers managed, and the garbage collection frequency settings. (Bounded disk usage that grows slowly in small increments)
/var/log/opsware	10 GB	The total log space used by all SA Core Components. (Fixed disk usage)
/var/opt/opsware	10 GB	The total run space used by all SA Core Components, including instances, pid files, lock files, and so on. (Fixed disk usage)

Table 7 SA Disk Space Requirements (cont'd)

SA Component Directory	Recommended Disk Space	Requirement Origin
/var/opt/opsware/word*	80 GB	The total disk space used by software that is imported into SA. Theoretically, this is infinite disk usage depending on how much software you import. Initial size calculation is based on the total size of all packages and patches that you want managed by SA. Known sizes range from 10GB to 250GB.
/var/opt/opsware/ogfs/mnt	20 GB	The home directory for the Global File System (OGFS) enabled SA user accounts.

* The entries in Table 7 marked with an asterisk are directory path defaults that you can change during the installation process. The recommended disk space for these directories is based on average-sized directories, which could be smaller or larger, according to usage.

For performance reasons, you should install the SA Components on a local disk, not on a network file server. However, for the Software Repository, you can use a variety of storage solutions, including internal storage, Network Attached Storage (NAS), and Storage Area Networks (SANs).

Model Repository (Database) Disk Space Requirements

Additional disk space is required for the Oracle software and the Model Repository data files. Keep in mind that storage requirements for the database grow as the number of managed servers grows.

As a benchmark figure, you should allow an additional 3.1 GB of database storage for every 1,000 servers in the facility that SA manages. When sizing the tablespaces, follow the general guidelines described in Table 8. If you need to determine a more precise tablespace sizing, contact your technical support representative.

Table 8 Tablespace Sizes

Tablespace	MB/1000 Servers	Minimum Size
AAA_DATA	256 MB	256 MB
AAA_INDX	256 MB	256 MB
AUDIT_DATA	256 MB	256 MB
AUDIT_INDX	256 MB	256 MB
LCREP_DATA	3,000 MB	1,500 MB
LCREP_INDX	1,600 MB	800 MB
TRUTH_DATA	1,300 MB	700 MB
TRUTH_INDX	400 MB	400 MB
STRG_DATA	1,300 MB	700 MB
STRG_INDX	400 MB	400 MB

Software Repository Disk Space Requirements

The Software Repository contains software packages and other installable files and is part of the *Slice Component bundle*. Typical installations start with approximately 300 GB allocated for the server hosting the Software Repository. However, more space might be required, depending on the number and size of the packages, as well as the frequency and duration of configuration backups.

Media Server Disk Space Requirements

Dependent on your OS Provisioning requirements. This component requires sufficient disk space for the OS media for all the operating system versions you intend to provision.

SA Core Performance Scalability

You can vertically scale the SA Core Components, by adding additional CPUs and memory, or horizontally, by distributing the Core Components to multiple servers.

Table 9 and Table 10 list the recommended distribution of SA components across multiple servers. In both tables, the bundled SA Core Components are distributed in the following way:

- MR: Model Repository
- INFRA: Infrastructure Component
  - Model Repository Multimaster Component
  - Management Gateway
  - Primary Data Access Engine
- Slice(*x*):
  - Agent Gateway
  - Core Gateway
  - Command Engine
  - Software Repository
  - Command Center
  - Build Manager
  - Web Services Data Access Engine
  - Secondary Data Access engine)
  - Global File System

Core Component Distribution

The introduction of bundled components requires that you consider how to distribute the SA Core components based on the hardware and memory you have available. A typical SA 7.5 installation now has three main components. The Model Repository, the Infrastructure Component bundle and one Slice Component bundle in addition to the Media Server and Boot Server. Since the Media Server and Boot Server do not generate much load and often have environmental dependencies they are not listed in the tables below.

There is no infallible way to select hardware for an SA installation. However, below are some recommended SA Core Component layouts that should perform well. As you can see, scaling a core requires adding slices. Each slice adds highly available UI, API, OGFS, Build Manager and Gateway resources. Consider that, when you have a small number of core servers, it may be best to begin with two larger servers, then grow the capacity of the core by adding additional slices. In Table 9 and Table 10, the following shorthand is used:

MR — Model Repository

INFRA — Infrastructure Component bundle

Slice <*X*> — Slice Component bundle

OS Prov — Operating System Provisioning Component bundle.

Table 9 Example Component Distribution: Two Large Servers and Additional Smaller Servers

Number of Managed Servers	**Number of Users**	**Number of Core Servers**	**SA Core Component Distribution by Server**				
			8 CPU Cores 8GB RAM	8 CPU Cores 8GB RAM	4 CPU Cores 8GB RAM	4 CPU Cores 8GB RAM	4 CPU Cores 8GB RAM
960	40	1	MR INFRA Slice 0 OS Prov				
2250	90	2	MR	INFRA Slice 0 OS Prov			
4500	180	3	MR	INFRA Slice 0 OS Prov	Slice 1		
7200	280	4	MR	INFRA Slice 0 OS Prov	Slice 1	Slice 2	
8000	300	5	MR	INFRA Slice 0 OS Prov	Slice 1	Slice 2	Slice 3

If your Oracle database deployment must run on four CPU cores due to licensing restriction, use Table 10.

Table 10 Example Core Component Distribution when Limited to Four CPU Cores

Number of Managed Servers	Number of Users	Number of Core Servers	SA Core Component Distribution by Server				
			4 CPU Cores 8GB RAM	4 CPU Cores 8GB RAM	4 CPU Cores 8GB RAM	4 CPU Cores 8GB RAM	4 CPU Cores 8GB RAM
480	20	1	MR INFRA Slice 0 OS Prov				
1125	45	2	MR	INFRA Slice 0 OS Prov			
2250	90	3	MR	INFRA Slice 0 OS Prov	Slice 1		
3600	144	4	MR	INFRA Slice 0 OS Prov	Slice 1	Slice 2	
4000	160	5	MR	INFRA Slice 0 OS Prov	Slice 1	Slice 2	Slice 3

Small Core Server Capacity

For small test/demonstration environments, the following single server core implementations are feasible. These configurations *are not* appropriate for production environments.

- 1 core server with 4 CPU cores, 8 GB RAM: 480 managed servers
- 1 core server with 2 CPU cores, 8 GB RAM: 150 managed servers

Factors Affecting Core Performance

The hardware requirements for SA vary based on these factors:

- The number of servers that SA manages
- The number and complexity of concurrent operations
- The number of concurrent users accessing the Command Center
- The number of facilities in which SA operates

Multimaster Mesh Scalability

To support global scalability, you can install an SA Core in each major facility, linking the cores in a Multimaster Mesh. The size of the SA Core in each facility can be scaled according to local requirements.

Multimaster Mesh Availability

In addition to Model Repository replication, a Multimaster Mesh supports the replication and caching of the packages stored in the Software Repository. Typically, the core in each facility owns the software that is uploaded to the core's Software Repository. To support availability, multiple copies of the packages can be maintained in remote Software Repositories. See the *SA Administration Guide* for more information.

The bundling of the Software Repository with the Slice Component bundle and the Software Repository Store with the Infrastructure Component bundle does not affect availability. The Software Repository reads the replicator configuration file to determine how to serve files from backed up directories.

Satellite Core CPU/Memory Requirements

Servers hosting SA Satellite Core installations must meet the following requirements:

- 2 CPUs per 1,500 managed servers per Satellite Core
- 2 GB RAM per 1,500 managed servers per Satellite Core

Load Balancing Additional Instances of Core Components

If SA must support a larger operational environment, you can improve performance by installing additional instances of the *Slice Component bundle* which provides you with these additional components per installation:

- Agent Gateway
- Core Gateway
- Command Center
- Software Repository
- Build Manager
- Web Services Data Access Engine
- Secondary Data Access engine
- Global File System

If you have installed multiple instances of the Slice Component bundle, load balancing between the instances occurs automatically as requests for load services are received by the Core Gateway. The Core Gateway handles incoming client connections and load balances them across the Slice Component bundles in the core.

You can also deploy a hardware load balancer for the servers that run additional instances of the Slice Component bundle. You can configure the load balancer for SSL session persistence (stickiness) with the least connections algorithm.

You can also put a load balancer in front of the Core Gateways, however, this will only load balance the Gateways, but with the added benefit that clients would have only one address to connect to and would failover gracefully in the event of a Slice Component bundle host failure.

Load Balancing does not affect validation of `httpProxy` certificates since the identity of the core is based on the address the clients use to connect, not the identity of the server that ultimately serves the request. All Slice Component bundles should be issued the same certificate and the hostname referenced in the certificate should match the DNS hostname that external clients use to connect. If a load balancer is used, this should be the hostname of the load balancer.

3 Pre-Installation Requirements

This section describes the system, environment, and network administration tasks that you must perform before you run the SA Installer.

Currently, any SA Core installations must be performed by HP Professional Services. HP cannot provide technical support for customer-performed SA Core installations. SA Satellite installations, however, can be performed by the customer.

Dual Layer DVD Requirements

The *SA Product Software DVD*, the *Oracle_SA DVD*, and the *SA Agent and Utilities DVD* require a dual layer DVD drive. See SA Installation Media on page 96 for information about the SA DVD set.

Solaris and Linux Requirements for Core Servers

This section describes platform-specific packages and utilities that must be installed for the operating system on the server that will host an SA Core.

The supported operating systems for SA Core Components are discussed in Chapter 2, “Operating System and Hardware Requirements.”

If you plan to install the Oracle database using the Oracle Universal Installer or use an existing Oracle installation rather than use the HP-supplied Oracle database, the server that hosts the Oracle RDBMS software (required by the Model Repository) has *additional* requirements, as described in Oracle Setup for the Model Repository on page 187.

Solaris Requirements

If you will be installing an SA Core Server under Solaris, you must ensure that the packages listed in Table 11are installed. Table 12 lists recommended packages and Table 13 lists packages that must *not* be installed.

Table 11 Packages Required for Solaris

Required Packages for Solaris		
SUNWCreq (cluster)	SUNWeurf	SUNWeudiv
SUNWadmap	SUNWi2rf	SUNWeudlg
SUNWadmc	SUNWi4rf	SUNWeudmg
SUNWdoc	SUNWi5rf	SUNWeuezt
SUNWesu	SUNWi7rf	SUNWeuhed
SUNWman	SUNWi8rf	SUNWeuluf
SUNWmkcdS	SUNWi9rf	SUNWeulux
SUNWswmt	SUNWi13rf	SUNWeuodf
SUNWtoo	SUNWi15rf	SUNWeuxwe
SUNWtoox**	SUNWtxfnt	SUNWuiu8
SUNWadmfw	SUNWinttf	SUNWuiu8x
SUNWlibC	SUNW5xmft	SUNWulcf
SUNWlibCx**	SUNWcxmft	SUNWulcfx
SUNWinst	SUNWjxmft	SUNWulocf
SUNWucbt	SUNWkxmft	SUNWuxlcf
SUNWucbtx**	SUNWeu8df	SUNWuxlcx
SUNWscpu	SUNWeu8os	SUNWeudbd
SUNWscpux**	SUNWeu8ox	SUNWeudhs
SUNWtcsh	SUNWeudba	SUNWeusru
SUNWsacom	SUNWeudda	SUNWuium
SUNWntpr	SUNWeudhr	NSCPeu8cm
SUNWntpu	SUNWeudis	
SUNWarrf		

** These packages are required only for Solaris 8 and Solaris 9.

Table 12 Packages Recommended for Solaris 8 and 9

Recommended Packages for Solaris		
SUNWisolc	SUNWi1of	SUNWiniu8
SUNWisolx	SUNWjiu8	SUNWiniu8x
SUNWislcc	SUNWjiu8	
SUNWislcx	SUNWkiu8	
SUNWciu8	SUNWkiu8x	
SUNWciu8x	SUNWtiu8	
SUNWhiu8	SUNWtiu8x	
SUNWhiu8x		

Table 13 Packages That Must Be Removed from Solaris

Packages That Must Me Removed From Solaris
SUNWCpm

Other Solaris Requirements

The SA Core Server must also meet the following requirements:

- On the server where you will install the SAS Web Client component, you must install the J2SE Cluster Patches for Solaris. To download these patches, search for "J2SE Cluster Patches" for your version of Solaris at *http://www.sun.com/*.
- On all core servers, verify that the Network File System (NFS) is configured and running.
- For Daylight Saving Time (DST) on Solaris 9 servers, you must install the time zone patch 113225-07 or later, and libc patch 112874-33 or later. To download these patches, search for the patch ID at *http://www.sun.com/*.
- For Daylight Saving Time (DST) on Solaris 10 servers, you must install the time zone patch 122032-03 or later, and libc patch 119689-07 or later. To download these patches, search for the patch ID at *http://www.sun.com/*.

  For more information about DST changes, search for "Daylight Saving Time (DST)" at *http://www.sun.com/*.

Linux Package Requirements

For Red Hat Linux AS 3 32-bit x_86, an SA Core Server must have the packages listed in Table 14 installed. For Red Hat Linux AS 4 32-bit x86 and Red Hat Linux Server 5 x_86, an SA Core Server must have the packages listed in Table 15 installed. For both and Red Hat Linux AS4 32-bit x86 and Red Hat Linux AS4 64-bit x86, the packages listed in Table 16 must *not* be installed.

Due to a known Linux AS4 64-bit x86 kernel bug, you must have Update 5 or later installed on all servers that will host an SA Core

Table 14 Required Packages For Linux As3 32-bit x_86

Required Packages	Architecture
at	32-bit x86
compat-db	32-bit x86
compat-libstdc++	32-bit x86
coreutils	32-bit x86
cpp	32-bit x86
expat	32-bit x86
gcc	32-bit x86
glibc-devel	32-bit x86
glibc-headers	32-bit x86
glibc-kernheaders	32-bit x86
iptables	32-bit x86
kernel-source	32-bit x86
libcap	32-bit x86

Table 14 Required Packages For Linux As3 32-bit x_86

Required Packages	Architecture
libxml2-python	32-bit x86
libstdc++	32-bit x86
libstdc++-devel **	32-bit x86
mkisofs *	32-bit x86
ncompress (contains uncompress utility)	32-bit x86
nfs-utils	32-bit x86
ntp	32-bit x86
patch	32-bit x86
patchutils	32-bit x86
sharutils	32-bit x86
strace	32-bit x86
unzip	32-bit x86
XFree86-libs	32-bit x86
XFree86-libs-data	32-bit x86
XFree86-Mesa-libGL	32-bit x86
xinetd	32-bit x86
zip	32-bit x86

* mkisofs is used for premastering ISO 9660 file systems used on CDROMs. It is open source and available at *http://freshmeat.net*, search for “mkisofs”.

** Required for Oracle database (Model Repository)

.

Table 15 Packages Required for Linux AS 4 x_64 and AS 5 x_64

Required Packages	Architecture
binutils	x86_64
chkfontpath	x86_64
compat-db	i386
compat-db	x86_64
cpp	x86_64
desktop-file-utils	x86_64
elfutils-libelf (Red Hat 5 only)	x86_64
elfutils-libelf-devel (Red Hat 5 only)	x86_64

Table 15 Packages Required for Linux AS 4 x_64 and AS 5 x_64 (cont'd)

Required Packages	Architecture
expat	i386
expat	x86_64
gamin-devel	x86_64
gcc	x86_64
gcc-c++	x86_64
glibc	i686
glibc	x86_64
glibc-common	x86_64
glibc-devel	i386
glibc-devel	x86_64
glibc-headers	x86_64
glibc-kernheaders (AS 4 only)	x86_64
iptables	x86_64
kernel (Red Hat 5 only)	x86_64
kernel-smp (AS 4 only)	x86_64
kernal-dev (Red Hat 5 only)	x86_64
kernel-smp-devel (AS 4 only)	x86_64
libaio	i386
libaio	x86_64
libcap	i386
libcap	x86_64
libgcc	i386
libgcc	x86_64
libpng	i386
libpng	x86_64
libpng10	i386
libpng10	x86_64
libstdc++	i386
libstdc++	x86_64
libstdc++-deve (Red Hat 5 only)	x86_64
libtermcap	i386

Table 15 Packages Required for Linux AS 4 x_64 and AS 5 x_64 (cont'd)

Required Packages	Architecture
libtermcap	x86_64
libxml2	i386
libxml2	x86_64
libxml2-python	x86_64
make	x86_64
mesa-libGL (Red Hat 5 only)	i386
mesa-libGL (Red Hat 5 only)	x86_64
mesa-libGLU (Red Hat 5 only)	i386
mesa-libGLU (Red Hat 5 only)	x86_64
mkisofs	x86_64
ncompress	x86_64
nfs-utils	x86_64
ntp	x86_64
openmotif (Red Hat 5 only)	NA
openmotif21 (AS 4 only)	NA
openmotif21	i386
patch	x86_64
patchutils	x86_64
pdksh	x86_64
popt	i386
popt	x86_64
readline	i386
readline	x86_64
rpm-build	x86_64
sharutils	x86_64
strace	x86_64
sysstat	x86_64
tcp_wrappers	i386
tcp_wrappers	x86_64
ttmkfdir	x86_64
unzip	x86_64

Table 15 Packages Required for Linux AS 4 x_64 and AS 5 x_64 (cont'd)

Required Packages	Architecture
vim-enhanced	x86_64
vnc	x86_64
vnc-server	x86_64
xinetd	x86_64
xinitrc	noarch
xorg-x11 (AS 4 only)	x86_64
xorg-x11-Mesa-libGL (AS 4 only)	i386
xorg-x11-Mesa-libGL	x86_64
xorg-x11-Mesa-libGLU (AS 4 only)	i386
xorg-x11-Mesa-libGLU (AS 4 only)	x86_64
xorg-x11-Xvfb (AS 4 only)	x86_64
xorg-x11-deprecated-libs (AS 4 only)	i386
xorg-x11-deprecated-libs (AS 4 only)	x86_64
xorg-x11-font-utils	x86_64
xorg-x11-libs	i386
xorg-x11-libs	x86_64
xorg-x11-xauth	x86_64
xorg-x11-xfs	x86_64
xterm	x86_64
zip	x86_64
zlib	i386
zlib	x86_64

Table 16 Packages That Must Be Removed for Linux

Packages		
samba	rsync	tftp (AS 3 and 4 only) **
apache	httpd	tftp-server (Red Hat 5 only)
yast2-dhcp-server (SLES 10 only)	yast2-samba-server (SLES 10 only)	dhcp**
	yast2-tftp-server (SLES 10 only)	

** Existing versions of the `tftp` and `dhcp` packages cannot reside on the same server as the OS Provisioning Boot Server component; however, they can reside on SA Core Servers that do not have the OS Provisioning Boot Server component.

To verify that the `samba` package, for example, is installed, enter the following command:

```
# rpm -qa | grep samba
```

You can obtain the latest versions of these packages from the Red Hat errata web site.

To remove packages, enter the following command:

```
# rpm -e package_name
```

Some packages in this list may be depended on by other packages that are installed on your system. For example, the default Red Hat installation includes `mod_python` and `mod_perl` that depend on `httpd` being installed. In order to remove packages that fulfill dependencies, you must simultaneously remove the packages that create the dependencies. In this example, you would need to enter the following command:

```
# rpm -e httpd mod_python mod_perl
```

If `rpm` identifies an additional dependency, it will note which packages have dependencies on the components to be removed and fail. These packages must be added to the uninstall command line. If the chain of dependencies cannot be suitably resolved, enter the `rpm -e --nodeps` command to remove the desired packages without considering dependencies.

Additional Linux Requirements

For Linux systems, you must also adhere to the following requirements:

- Red Hat Enterprise Linux 4 AS must be at least Update 5.
- You must specify the server's initial run level as level 3 in the `/etc/inittab` file.
- If the server uses Integrated Drive Electronics (IDE) hard disks, you must enable Direct Memory Access (DMA) and some other advanced hard disk features that improve performance by running the following script as `root` on the server and then reboot the server:

  ```
  # cat > /etc/sysconfig/harddisks << EOF
  USE_DMA=1
  MULTIPLE_IO=16
  EIDE_32BIT=3
  LOOKAHEAD=1
  EOF
  ```

- Due to a bug in the Linux kernel, you must configure the loopback interface to use a Maximum Transmission Unit (MTU) size of 16036 bytes or less. To make this change, perform the following tasks:

  a. Run the `ifconfig lo mtu 16036` command. This sets the MTU of the running kernel.

  b. Add the line `MTU=16036` to the end of the `/etc/sysconfig/network-scripts/ifcfg-lo` file. This causes the MTU to be properly set when the system is booted.

- Disable the Security-Enhanced Linux kernel (SELinux) on all core servers running Linux AS4 64-bit x86.
- For Daylight Saving Time (DST) on Red Hat Enterprise Linux AS 3 and AS 4, you must install the latest time zone data. You can download these time zone updates from the following location:

  https://rhn.redhat.com/errata/RHEA-2006-0745.html

- For Daylight Saving Time (DST) on SUSE Linux Enterprise Server 9, you must install the latest time zone data. You can download these updates from the following location:

http://www.novell.com/support/

- For Daylight Saving Time (DST) on Sun Solaris, you must install the latest time zone data. You can download these updates from *http://www.sun.com*.
- If you are using a Linux NFS server, be aware that, by default, Linux enables NFSv3, which prevents Solaris servers from entering the server pool. You can either disable NFSv3 on the Linux NFS server or you can add DHCP options to force Solaris 10 to use NFSv2:
  - To force the Solaris `miniroot` to use NFSv2, add the following lines to your DHCP configuration file:
    - In the generic section of the DHCP configuration file, add the following lines:

      ```
      # added for nfs 2 miniroot
      option SUNW.SrootOpt code 1 = text;
      # end of nfs 2 miniroot stuff
      ```

    - In the `solaris-sun4u`, `solaris-sun4us`, and solaris-specific-kernel classes, add the following lines:

      ```
      # added for nfs 2 miniroot
      option SUNW.SrootOpt "vers=2";
      # end of nfs 2 miniroot stuff
      ```

  - To disable NFSv3 on the Linux NFS server add the following lines to the `/etc/sysconfig/nfs` file and then restart NFS:

    ```
    MOUNTD_NFS_V3=no
    MOUNTD_NFS_V2=yes
    RPCNFSDARGS='--no-nfs-version 4'
    ```

SUSE Linux Enterprise Server 10 Package Requirements

For SUSE Linux Enterprise Server 10 64-bit x_86, an SA Core Server must have the packages listed in Table 17 installed.

Table 17 SUSE Linux Enterprise Server 10 Required Packages

Required Packages	Architecture
binutils	x86_64
cpp	x86_64
desktop-file-utils	x86_64
expat	x86_64
gcc-c++	x86_64
gcc	x86_64
glibc	x86_64
glibc-32bit	x86_64
glibc-devel	x86_64
glibc-devel-32bit	x86_64
iptables	x86_64

Table 17 SUSE Linux Enterprise Server 10 Required Packages

Required Packages	Architecture
kernel-smp	x86_64
kernel-source	x86_64
libaio	x86_64
libaio-32bit	x86_64
libaio-devel	x86_64
libcap	x86_64
libcap-32bit	x86_64
libelf	x86_64
libgcc	x86_64
libstdc++	x86_64
libstdc++-devel	x86_64
libpng	x86_64
libpng-32bit	x86_64
libxml2	x86_64
libxml2-32bit	x86_64
libxml2-python	x86_64
make	x86_64
mDNSResponder-lib	x86_64
mkisofs	x86_64
ncompress	x86_64
nfs-utils	x86_64
patch	x86_64
popt	x86_64
popt-32bit	x86_64
readline	x86_64
readline-32bit	x86_64
rpm	x86_64
sharutils	x86_64
strace	x86_64
sysstat	x86_64
termcap	x86_64

Table 17 SUSE Linux Enterprise Server 10 Required Packages

Required Packages	Architecture
unzip	x86_64
vim	x86_64
xinetd	x86_64
xntp	x86_64
xorg-x11-libs	x86_64
xorg-x11-libs-32bit	x86_64
xorg-x11	x86_64
xterm	x86_64
zip	x86_64
zlib	x86_64
zlib-32bit	x86_64

Requirements for Installing Oracle 11g using the SA Installer

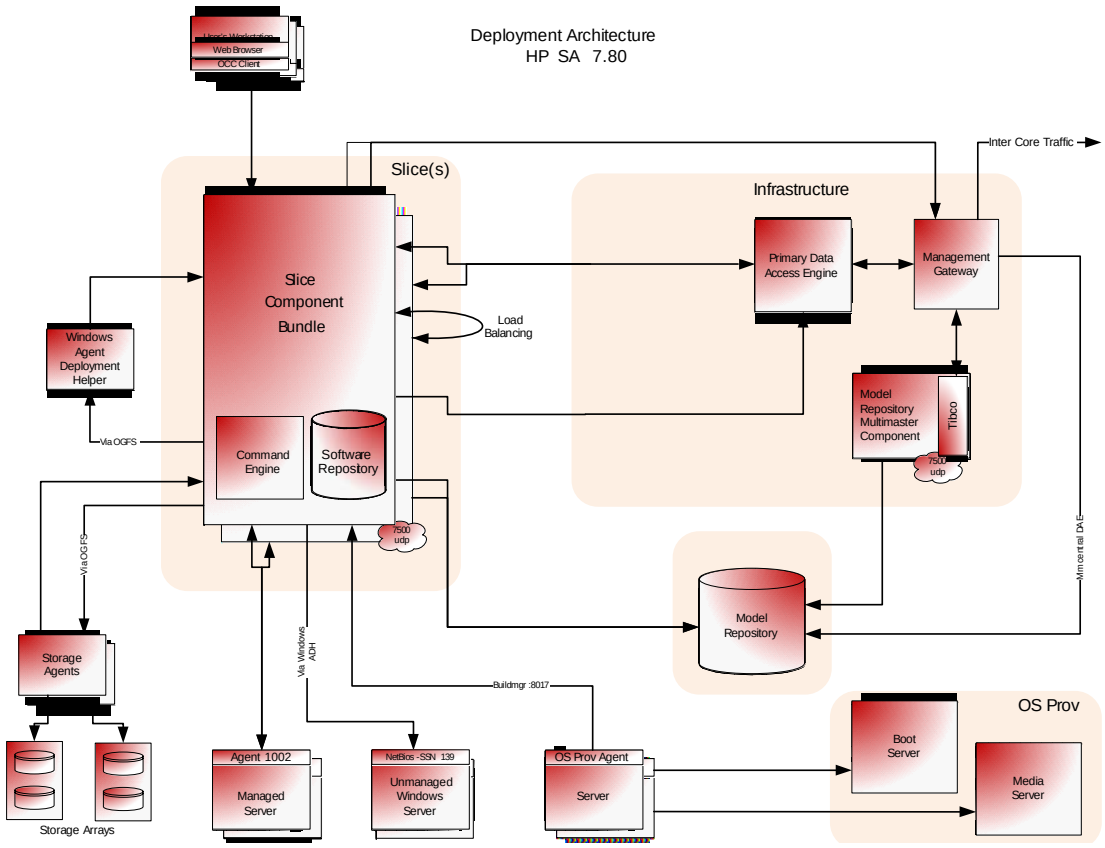
The Model Repository requires an installed Oracle database. You can use the SA Installer to install the HP-supplied Oracle 11g database on a Solaris 10 x86_64 server or on a Red Hat Enterprise Linux 4 AS x86_64, Red Hat Enterprise Linux 5 AS x86_64, or SUSE Linux Enterprise Server 10 x86_64 server. You can also use a pre-existing Oracle installation. Whatever method you choose, you should see Oracle Setup for the Model Repository on page 187 for more information.

Network Requirements

This section discusses the network requirements within a facility, open ports required for Core Components, and name resolution requirements. These requirements must be met for both First Cores, Multimaster Mesh installations, and Satellite cores.

Figure 11 shows the network layout for an SA Single Core configuration.

Figure 11 Network Layout for a Single SA Core



Network Requirements within a Facility

Before running the Installer, your network environment must meet the following requirements:

- All SA Core Servers must be on the same Local Area Network (LAN or VLAN).
- There must be full network connectivity between all SA Core Servers and the servers that the SA Core will manage.
- Core Servers expect user accounts to be managed locally and cannot use the Network Information Service (NIS) directory to retrieve password and group information. During installation of the Core Components, the installer checks for the existence of certain target accounts before creating them. If you are using NIS, this check will fail.
- If you plan to use network storage for Core Components, such as the Software Repository or OS Provisioning Media Server, you must ensure that the `root` user has write access over NFS to the directories where the components will be installed.

- The speed and duplex mode of the Core's and Managed Servers' NIC adapters must match the switch they are connected to. A mismatch will cause poor network performance between the Core and Managed Servers.

Open Ports

You must configure any firewalls protecting your Core Servers to allow the ports shown in Table 18 to be open. Note that the ports numbers listed in the table are the default values which can be changed during the installation, so ensure you are leaving the correct ports open.

Table 18 Open Ports on a Firewall Protecting an SA Core

Port	Component	Purpose
80 (TCP)	Command Center	HTTP redirector
443 (TCP)	Command Center	HTTPS Proxy for SAS Web Client UI, SAS Client, SA Web Services (2.2)
1003 (TCP)	Software Repository (`word`)	Core Communicatons
1004 (TCP)	Data Access Engine (`spin`)	Core Communicatons
1018 (TCP)	Command Engine (`way`)	Core Communicatons
1032 (TCP)	Web Services Data Access Engine (`twist`)	Core Communicatons
1521 (TCP)	Model Repository (`truth`)	Database Communicatons
2001 (TCP)	Management Gateway/ Core Gateways	Inbound tunnels from other Gateways (If Port 2001 is in use, rolls over to 2003)
2222 (TCP)	Global File System	Global shell session from an SSH client
8017 (UDP, TCP)	Agent Gateway	Interface to the Build Manager
8080 (TCP)	Command Center	Load Balancing Gateway for the SAS Client

Table 19 shows the ports used by the OS Provisioning components that are accessed by servers during the provisioning process. (In SA, OS provisioning refers to the installation of an operating system on a server.)

Table 19 Open Ports for the OS Provisioning Components

Port	Component	Service
67 (UDP)	Boot Server	DHCP
69 (UDP)	Boot Server	TFTP
111 (UDP, TCP)	Boot Server, Media Server	RPC (`portmapper`), required for NFS
Dynamic/Static*	Boot Server, Media Server	`rpc.mountd`, required for NFS

Table 19 Open Ports for the OS Provisioning Components (cont'd)

Port	Component	Service
2049 (UDP, TCP)	Boot Server, Media Server	NFS
8017 (UDP, TCP)	Agent Gateway	Interface to the Build Manager
137 (UDP)	Media Server	SMB NetBIOS Name Service
138 (UDP)	Media Server	SMB NetBIOS Datagram Service
139 (TCP)	Media Server	NetBIOS Session Service
445 (TCP)	Media Server	MS Directory Service

* By default, the `rpc.mountd` process uses a dynamic port, but it can be configured to use a static port. If you are using a dynamic port, the firewall must be an application layer firewall that can understand RPC requests that clients use to locate the port for `mountd`.

The OS Provisioning Boot Server and Media Server run various services (such as `portmapper` and rpc.mountd) that could be susceptible to network attacks. It is recommended that you segregrate the OS Provisioning Boot Server and Media Server components onto their own DMZ network. When you segregate these components, the ports listed in Table 19 should be opened to the DMZ network from the installation client network. Additionally, the Boot Server and Media Server should have all vendor-recommended security patches applied.

Table 20 shows the Managed Server port that must be open for SA Core Server connections.

Table 20 Open Ports on Managed Servers

Port	Component
1002 (TCP)	SA Agent

Host and Service Name Resolution Requirements

SA must be able to resolve Core Server host names and service names to IP addresses through proper configuration of DNS or the `/etc/hosts` file.

Previous Releases

If you plan to install the Core Components on a server that had a previous SA installation (for example, version 6.x or 7.0), you must verify that the host names and service names resolve correctly for the new installation.

Core Servers and Host/Service Name Resolution

During the installation, the `/etc/hosts` file on machines where the *Slice Component bundle* is installed will be modified to contain entries pointing to the *Secondary Data Access Engine*, the *Command Center*, the *Build Manager*, and the fully qualified domain name of the `localhost`.

All other servers hosting Core Components must be able to resolve their own valid host name and the valid host name of any other SA Core Server (if you will be using a multiple core installation or Multimaster Mesh). A fully qualified name includes the subdomain, for example, `myhost.acct.buzzcorp.com`. Enter the `hostname` command and verify that it displays the fully qualified name found in the local `/etc/hosts` file.

Additionally, a Core Server must be able to resolve both the fully qualified and unqualified names of the SA Services. (Each service name represents an SA Core Component.) For example, both `truth` (unqualified) and `truth.acct.buzzcorp.com` (fully qualified) must resolve to the IP address of the server containing the Model Repository.

The list of fully qualified names of the SA services follows:

- `truth.`*`subdomain`* — Model Repository
- `way.`*`subdomain`* — Command Engine
- `spin.`*`subdomain`* — Primary Data Access Engine
- `theword.`*`subdomain`* — Software Repository
- `wordcache.`*`subdomain`* — Software Repository Multimaster Component The name `wordcache` must resolve to the core server running the Software Repository (Slice Component bundle).

In a *typical* component layout, the Software Repository Store is installed as part of the Infrastructure Component bundle and the Slice Component bundle must able to map the IP of the Infrastructure host to its hostname. In a *custom* component layout, the Software Repository Store may be installed separately on any host, therefore the Slice Component bundle must be able to map the IP of that host to its hostname. It is a common practice, but not a requirement, to host the Software Repository Store and the `OGFS home/audit` directories on the same server.

On Solaris 10, an OGFS installation requires the actual host name of the OGFS host. In the `dfstab` file on the Software Repository host, specify the actual hostname of the OGFS host.

OS Provisioning: DHCP Proxying

If you plan to install your OS Provisioning components on a separate network from the Core Components, you must set up DHCP proxying to the DHCP server (for example, using Cisco IP Helper). If you use DHCP proxying, the server/router performing the DHCP proxying must also be the network router so that PXE can function correctly.

The OS Provisioning Boot Server component provides a DHCP server, but does not include a DHCP proxy. For DHCP server configuration information, see DHCP Configuration for OS Provisioning on page 127.

Windows Patch Management Requirements

The SA Windows Patch Management feature requires that, before running the Installer, you obtain several files from the Microsoft software download repository and copy them to a directory that will be accessible during the SA installation. During the installation process, the Installer will prompt you to enter the fully qualified path to the Microsoft files in this directory and will fail if the files do not exist at the specified location.

Supported Platforms

- Windows 2000
- Windows XP
- Windows Server 2003 x86 and x64
- Windows Server 2008 x86 and x64
- Windows Server 2008 x86 Server Core and Windows 2008 x64 Server Core

In order to apply patches to Managed Servers running Windows Server 2000 SP4 and Windows Server 2003 RTM, you must first ensure that the Microsoft update MS04-011 (or a subsequent update) has been applied to those servers. This update is required for MBSA 2.1 to run properly.

Requirements

The Managed Servers meet the following Windows patching requirements:

- Windows Installer 3.1 must be installed
- MSXML 3+ must be installed
- The Windows Update Agent must be installed
- The Windows (Automatic) Update service must *not* be disabled but must be set to *never* check for updates.

As of Windows Server 2008, the Automatic Update service was renamed the Windows Update service.

Installing MBSA 2.1 for SA 7.80

To obtain the required Windows patch management files, perform the following tasks:

1. Obtain the following files from Microsoft:
   - qchain.exe

     The `qchain.exe` utility is a command-line program that chains hotfixes together. When you chain updates, you install multiple updates without restarting the computer between each installation.

To download the package containing `qchain.exe`, search for "`qchain.exe`" at *http://www.microsoft.com* . Install the package on a Windows machine and note the location of the `qchain.exe` file.

- wsusscn2.cab

  The `wsusscn2.cab` file contains the Microsoft patch database. To download the package containing `wsusscn2.cab`, search for "`wsusscn2.cab`" at *http://www.microsoft.com* .

- WindowsUpdateAgent-x86.exe

  The `WindowsUpdateAgent30-x86.exe` file is required by the `mbsacli.exe` utility. To download the package containing `WindowsUpdateAgent30-x86.exe`, search for "Windows Update Agent" at *http://www.microsoft.com*. After downloading, you must rename the file "`WindowsUpdateAgent-x86.exe`".

- WindowsUpdateAgent-x64.exe

  The `WindowsUpdateAgent30-x64.exe` file is required by the `mbsacli.exe` utility. To download the package containing `WindowsUpdateAgent30-x64.exe`, search for "Windows Update Agent" at *http://www.microsoft.com*. After downloading, you must rename the file "`WindowsUpdateAgent-x64.exe`".

- WindowsUpdateAgent-ia64.exe

  The `WindowsUpdateAgent30-ia64.exe` file is required by the `mbsacli.exe` utility. To download the package containing `WindowsUpdateAgent30-ia64.exe`, search for "Windows Update Agent" at *http://www.microsoft.com*. After downloading, you must rename the file "`WindowsUpdateAgent-ia64.exe`".

- mbsacli.exe (version 2.1)

  This file is packaged with the MBSA 2.1 setup file, `MBSASetup-x86-EN.msi`, that you must download by searching for "`MBSA 2.1`" at *http://www.microsoft.com* .

  After the download, on a Windows machine run `MBSASetup-x86-EN.msi` to install MBSA 2.1. In the directory where you installed MBSA 2.1, locate the `mbsacli.exe` file. By default, the file is installed here:

  ```
  %program files%\Microsoft Baseline Security
  Analyzer 2\mbsacli.exe
  ```

- wusscan.dll

  The `wusscan.dll` file is in the directory where you installed MBSA 2.1. By default, the file is here:

  ```
  %program files%\Microsoft Baseline Security
  Analyzer 2\wusscan.dll
  ```

2. Copy the files you obtained in the preceding steps to a directory that will be accessible by the SA Installer during the Software Repository installation. For example, you might copy the files to the following directory:

   ```
   /opsw/win_util
   ```

3. Verify that the destination directory contains all these files:

```
mbsacli.exe
WindowsUpdateAgent-x86.exe
WindowsUpdateAgent-x64.exe
WindowsUpdateAgent-ia64
qchain.exe
wsusscn2.cab
wusscan.dll
```

4 Write down the name of the directory containing the files. When you run the Installer, during the Software Repository installation, you will prompted to provide the fully qualified directory path. The location you provide will be stored in the parameter, `windows_util_loc`.

These patch management files will be copied to Windows servers during SA Agent deployment. If you upload newer versions of the files to the Software Repository later, they will be downloaded to the managed Windows servers during software registration. After the core is installed and running, you can upload new versions of these files with the Patch Settings window of the SAS Client. For more information, see the *SA Planning and Installation Guide*.

For information on Windows Patch Management, see the *SA User's Guide: Application Automation*.

Configuration Tracking Requirements

The Configuration Tracking feature tracks, backs up, and recovers critical software and system configuration information across Unix and Windows servers. When you enable the SA Configuration Tracking feature for a facility, by default, a separate partition is created on the server running the Software Repository. That partition will contain this Configuration Tracking backup directory:

`/var/opt/opsware/word/<facility-name>/acsbar`

You can optionally specify that the backup directory be created under the Software Repository root directory during SA installation.

The Configuration Tracking feature uses this directory to store backups of tracked configuration files and databases. The configuration tracking backup directory is relative to the Software Repository root directory:

`<word_root>/<facility_name>/acsbar`

Global File System (OGFS) Requirements

This section discusses requirements for the Global File System (OGFS).

OGFS Store and Audit Hosts

When you run the SA Installer interviewer in advanced mode, you can specify values for the `ogfs.store.host.ip` and `ogfs.audit.host.ip` parameters. (See Global File System Prompts on page 94.) If you set either of these parameters to point to a host that does not run the Slice Component bundle (which contains OGFS and the Software repository), then perform the following steps on the host you do specify:

1 With `mkdir`, create the directories that you specified for the `ogfs.store.path` and `ogfs.audit.path` parameters.

2 Modify the export tables.

In these examples, the Slice Component bundle is installed on two separate hosts within the same core.

a On a Solaris host, modify the `/etc/dfs/dfstab` file, similar to this:

```
# Begin Opsware ogfs export
share -F nfs -o anon=0,rw=1.2.3.4:1.2.3.5 /export/ogfs/store
share -F nfs -o anon=0,rw=1.2.3.4:1.2.3.5 /export/ogfs/audit
# End Opsware ogfs exports
```

where 1.2.3.4 and 1.2.3.5 are example IP addresses of the two Slice Component bundle hosts and where `/export/ogfs/store` and `/export/ogfs/audit` are corresponding paths that exist on the host from where you are exporting the OGFS data.

b On a Linux host, modify the `/etc/exports` file, such as:

```
# Begin Opsware ogfs export
/export/ogfs/store 1.2.3.4(rw,no_root_squash,sync) \
1.2.3.5(rw,no_root_squash,sync)
/export/ogfs/audit 1.2.3.4(rw,no_root_squash,sync) \
1.2.3.5(rw,no_root_squash,sync)
# End Opsware ogfs exports
```

where 1.2.3.4 and 1.2.3.5 are example IP addresses of the two Slice Component bundle hosts and where `/export/ogfs/store` and `/export/ogfs/audit` are corresponding paths that exist on the host from where you are exporting the OGFS data.

3 After you add new entries to the export tables, export the directories or restart the Network File System using standard system procedures.

Remember to verify that the NFS Daemon will start when the system reboots.

Name Service Caching Daemon (nscd) and OGFS

If the Name Service Caching Daemon (`nscd`) runs on the same server as the Slice Component bundle, then users cannot open a global shell session with a direct `ssh` connection. If `ncsd` is running on the Slice Component bundle server, the Installer turns it off and runs the `chkconfig nscd off` command to prevent it from starting after a reboot. No action is required.

Time and Locale Requirements

This section discusses the time and locale requirements for SA Core Servers.

Core Time Requirements

Core Servers (either Single Core or Multimaster) and Satellite Core Servers must meet the following requirements. These time requirements do not apply to Managed Servers.

- All SA Core Servers must have their time zone set to Coordinated Universal Time (UTC).
- All SA Core Servers must maintain synchronized system clocks. Typically, you will synchronize the system clocks through an external server that uses NTP (Network Time Protocol) services.

Linux Time Configuration

To configure the time zone on a Linux server, perform the following tasks:

1 Copy or link

```
/usr/share/zoneinfo/UTC
```

to

```
/etc/localtime.
```

2 Ensure that the `/etc/sysconfig/clock` file contains the following lines:

```
ZONE="UTC"
UTC=true
```

Solaris Time Configuration

To configure the time zone on a Solaris server, verify that the `/etc/TIMEZONE` file contains the following line:

```
TZ=UTC
```

Locale Requirements

The servers hosting the Model Repository and the Software Repository (part of the Slice Component bundle) must have the `en_US.UTF-8` locale installed.

To display data from Managed Servers using various locales, the server hosting the Global File System (OGFS) must also have all the locales installed.

To enable non-English locales for Windows patching, follow the instructions in "Locales for Windows Patching" in the *SA User's Guide: Application Automation*.

To verify whether the `en_US.UTF-8` locale is installed on a server, enter the following command:

```
echo $LANG
```

To define or modify the locale, enter the following values in the `/etc/sysconfig/i18n` file:

```
LANG="en_US.UTF-8"
SUPPORTED="en_US.UTF-8:en_US:en"
```

User and Group Requirements For Solaris and Linux

During installation on Solaris and Linux servers, the SA Installer creates two users (if you are installing OMDB, its installer will also add a user).

For Solaris, these users and groups are:

Table 21 Users and Groups Created During an SA/Solaris Install

userid	group	uid	groupid	home directory	shell
twist	twist		other	/var/opt/opsware/twist	/bin/sh twist
occ	occ		occ	/var/opt/opsware/occ	/bin/sh occ

For Linux, these users and groups are:

Table 22 Users and Groups Created During an SA/Linux Install

userid	group	uid	groupid	home directory	shell
twist	twist		users	/var/opt/opsware/twist	/bin/sh twist
occ	occ		occ	/var/opt/opsware/occ	/bin/sh occ

If your security policies disallow the creation of these users and groups during installation, you will need to add them manually.

4 Installation Methods and Checklists

The section reviews the types of SA installations, gives a general outline of the core installation process, and provides checklists that will help you prepare for and complete the installation process.

Types of SA Installations

There are three basic types of HP Server Automation installations: First Core (Single Core), Multimaster Mesh, and Satellite.

- **First Core or Single Core** (formerly Standalone Core): A Single Core typically provides management capabilities for servers in a single facility. By definition, a Single Core does not communicate or exchange information with other SA Cores however it can communicate with Satellite installations in remote facilities. The core contains all SA components including the Management Gateway, so it can easily become the First Core for a Multimaster Mesh.
- **Multimaster Mesh**: A *Multimaster Mesh* is a set of two or more SA Cores that communicate through Management Gateways and can perform real-time synchronization of the data about their Managed Servers contained in their respective Model Repositories over the network.

  The Model Repositories in each of the cores are continually updated so that they are always exact duplicates of each other. All the servers in a Multimaster Mesh can be managed through a single Command Center. A Multimaster Mesh is best for larger networks that span multiple facilities.
- **Satellite**: Satellite installations are appropriate for smaller, remote sites that may not have the installed infrastructure for a full core installation. A satellite installation is not a full core installation (it does not include the Model Repository), but it does provide some core capabilities by providing network communication with a core through a gateway. It also allows you to manage bandwidth between connected sites. A Satellite installation must be linked to at least one core, which can be either a Single Core or part of a Multimaster Mesh.

This guide uses the term *facility* to refer to the collection of servers and devices that reside in a single physical location. A facility can be all or part of a data center, server room, or computer lab. Each SA Core or Satellite is associated with a specific facility.

SA Core Installation Process Flow

The six main phases of the SA core installation process are summarized below. For more detailed information, see the cross references associated with each step.

1. **Planning**: In the planning phase, you must decide which facilities and servers you will manage with SA. You must also choose the type of SA installation that is appropriate for your site(s) and ensure that you have the required hardware and software, including operating systems, and sufficient network connectivity.

   See Chapter 1, "SA Architecture"

   See Chapter 2, "Operating System and Hardware Requirements" on page 33 of this guide for more information.

2. **Pre-installation Requirements**: Before beginning a core installation, whether it is a Single Core or a core in a Multimaster Mesh, you must perform such administrative tasks as ensuring that host names can be resolved, required ports are open and available, and installing any necessary operating system utilities, packages, and/or patches.

   See Chapter 3, "Pre-Installation Requirements" on page 47 of this guide for more information.

3. **Prerequisite Information for Installer Interview**: the SA Installer Interview Mode requires that you have certain information about your operational environment available because you will be asked to enter it during the interview. The information you provide will be saved into a *Response File* that will be used to set up the Core Server environment. You must gather this information and have it at hand as you run the pre-installation interview. Some examples of the information required are the name of the Facility to be managed by the core, the authorization domain, host names and IP addresses, and passwords used for SA users and the Oracle database, and so on.

   For a detailed description of the information required during the Installer Interview, see Chapter 5, "Prerequisites for the Installer Interview".

4. **SA Core Installation**: During this phase, you will run the Installer, complete the pre-installation interview to create the required response file, and then install one of the following types of Cores or Satellites:

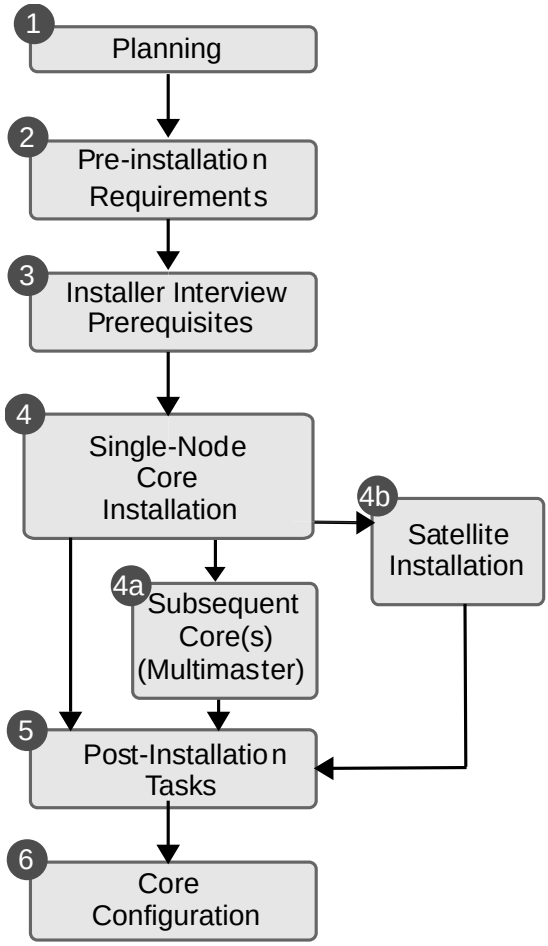
   - **First or Single Core Installation**: See Chapter 6, "Installing the First Core" on page 103 of this guide for more information.
   - **Subsequent Core Installations for a Multimaster Mesh:** See Chapter 8, "Multimaster Mesh Installation" on page 141 of this guide.
   - **Satellite Core Installation**: See Chapter 9, "Satellite Installation" on page 157 of this guide for more information.

5. **Post-installation Tasks**: See Chapter 7, "First Core Post-Installation Tasks" on page 115 of this guide.

6. **Core Configuration**: You will configure SA, performing tasks such as creating SA users and groups. At the end of this phase, SA is ready for operational use by system administrators. See the *SA Administration Guide* for more information.

Figure 12 shows the overall process of an SA core installation.

Figure 12 SA Core Installation Process Flow



Installation Checklists

This section provides the following pre-installation checklists that you may find helpful in planning your SA installation:

- Overall Planning Checklist
- Core-Specific Planning Checklist
- Specific Core Requirements Checklist
- Pre-Installation Tasks Checklist
- Post-Installation Tasks Checklist

Overall Planning Checklist

The following checklist summarizes decisions regarding the overall design of your SA installation.

Table 23 Overall Planning Checklist

Overall Planning Item	Answer
How many Facilities (locations/data centers) will you manage with SA?	
In each of these Facilities, how many servers do you expect to manage with SA?	
What is the naming convention for the Facilities? (For example, you might use site, building, or city names.)	
Have you taken an inventory of the operating systems and applications on the servers that you will manage with SA?	
Are all installed operating systems on the servers you plan to manage compatible with SA?	
Which of the following SA architectures have you chosen? — Single Core/First Core — Multimaster Mesh	
Do you plan to install Satellites?	
Which SA features will you use?	
What is your installation schedule for the SA Core and its components and for deploying SA Agents on the servers to be managed?	
Will you install the OS Provisioning Boot Server/Media Server bundle?	
Which operating systems will you provision (install) with OS Provisioning?	
What applications will you provision (install) with SA?	

Table 23 Overall Planning Checklist (cont'd)

Overall Planning Item	Answer
If you will be using Multimaster capabilities, how fast are the network connections between the SA Cores?	
Will you need Satellite capabilities, if so for how many sites?	
How many servers will be managed by the Satellite?	
In which remote Facilities will you install Satellite Cores?	
With which Cores will the Satellite communicate?	
How fast are the network connections between the Satellites and the core?	
Have you diagrammed your system including the hosts that will run the SA Core Components? If applicable, the diagram should show the network connectivity between Multimaster Cores and between Cores and Satellites.	

Core-Specific Planning Checklist

The following checklist of design decisions should be completed for each SA Core installation.

Table 24 Core-Specific Planning Checklist

Specific Core Planning Item	Answer
What is the name of the facility that this core will be associated with?	
For the Primary (First) Core, what is the Facility ID and the default customer name?	
How many servers will this Core manage?	
Will the Model Repository use: • The default Oracle software and database installed by the SA Installer? • An existing Oracle installation? Who is the DBA? Have you contacted the DBA about the required Oracle configuration changes needed for SA?	
Will you distribute the Core Components across multiple servers? If yes, diagram where the components are to be installed.	
What are the host names of the servers on which the Core Components will be installed?	
For a multiple-server core, will you have multiple instances of the Slice Component bundle?	

Table 24 Core-Specific Planning Checklist (cont'd)

Specific Core Planning Item	Answer
Will you install the following components into their own DMZ network? • OS Provisioning Boot Server • OS Provisioning Media Server	
Do you have the necessary licenses for Oracle?	
Have you written your backup and recovery plan for the servers running SA?	
Have you contacted your database administrator (DBA)? Your DBA will need to monitor the Oracle database when it goes into production.	
Have you contacted your network administrator? the network administrator will need to setup host name resolution (`/etc/hosts`, DNS) before the installation and may need to run a DHCP configuration tool after the installation.	

Specific Core Requirements Checklist

The following checklist summarizes the technical requirements that must be met on each server before each SA Core installation.

Table 25 Specific Core Requirements Checklist

Requirement	Answer
Have the servers on which you will install the Core Components been racked and stacked?	
Do you have root access to these servers?	
Do you have the permissions required to mount the SA DVDs and copy their contents to the Core Servers?	
Are the Core Servers running a supported operating system?	
Do the Core Servers meet the minimum CPU requirements?	
Do the Core Servers meet the minimum memory requirements?	
Do the Core Servers meet the disk space requirements?	
Are the servers for an individual core on the same LAN or VLAN? (Multimaster Cores must be on separate VLANs.)	
Do the Core Servers have network connectivity to the servers they will manage?	
Have you verified that Network Information System (NIS) is *not* running on the Core Servers?	

Table 25 Specific Core Requirements Checklist (cont'd)

Requirement	Answer
If you will be using the Network File System (NFS) for Core Components, such as the Software Repository or Media Server, does the root user have write access over NFS to the directories where the components are to be installed?	
Does the link speed and duplex of the core and managed servers match the switch to which they are connected?	
Are the necessary TCP ports open on the core and managed servers?	

Pre-Installation Tasks Checklist

The following checklist summarizes the tasks you must perform before installing an SA Core.

Table 26 Pre-Installation Tasks Checklist

Pre-installation Task	task completed?
For the servers that will run the Core Components, perform the specific tasks for Linux and Solaris described in the section Solaris and Linux Requirements for Core Servers on page 47.	
Set up the host name resolution (`/etc/hosts` or DNS) for the core servers.	
If OS Provisioning occurs on a separate network from the Core Components, set up DHCP proxying.	
Obtain `mbsacli.exe` and the other utilities required for patches from Microsoft and copy them to a location on your network that is accessible by the installer.	
Synchronize the system clocks on the Core Servers with an external Network Time Protocol (NTP) service.	
For a Multimaster Mesh installation, see the section Prerequisites for Multimaster Mesh Installations on page 142.	
Verify that you have followed the instructions in Chapter 5, "Prerequisites for the Installer Interview".	

Post-Installation Tasks Checklist

The following checklist summarizes the tasks you must perform *after* installing an SA core. For more information, see the "Post-Installation Tasks" chapter of the *Opsware® SAS Planning and Installation Guide*.

Table 27 Post-Installation Tasks Checklist

Post-installation Task	task completed?
Install the Windows Agent Deployment Helper.	
OS Provisioning: Configure DHCP for OS Provisioning. You may use the DHCP server included with SA or an external DHCP server.	
OS Provisioning: For Windows OS provisioning, the host name `buildmgr` should resolve on Windows installation clients.	
Patch Management: (on Windows NT or 2000) Create a silent-installable version of IE 6.0 or later.	
Multimaster Mesh: Associate customers with the new facility.	
Multimaster Mesh: Update the group permissions for the new facility.	
Multimaster Mesh: Verify that the multimaster transaction traffic is flowing between the cores.	

5 Prerequisites for the Installer Interview

This section lists the information about your environment that you will need gather to complete the SA Installer interview. It also provides information about the installer command line (CLI) syntax, log files, and SA Installer distribution on DVDs.

The SA Installer Interview Mode

Before installing the First Core, you must run the Installer in Interview Mode to provide certain information about your facility's environment. For example:

- Passwords (SA Administrator, Database Administrator, etc.)
- Service Names (TNS name)
- Configuration parameter values
- Path names for programs, configuration file, logs
- IP Addresses for Core hosts and devices hosting Core Components
- Gateway port numbers, etc.

When started in *Interview Mode*, the Installer displays a series of *prompts* to which you will provide information about your environment. The specific prompts will vary depending on whether you choose the *Simple* or *Advanced* interview mode. All responses you make will be stored on the server from which you run the Installer in a *Response File* that is used during the installation.

After the interview completes, you can either continue the installation using the response file you just created or quit and continue the installation later. You may also use this response file when you install Subsequent Cores in a Multimaster Mesh or do a core upgrade. Therefore, you should record the location of the response file so that it can be easily found.

SA Installer Interview Prompts

Before you run the Installer interview, you must gather the information that you will enter when prompted during the interview process. Examples of this information are: the password for the Oracle `opsware_admin` user, the Facility name for the core, and the SA authorization domain, etc.

Use the tables below, which list the various prompts that you will see when running the Installer interview, to compile your responses before invoking the Installer Interview. Prompts seen only during the Advanced Interview mode are labeled with the word **Advanced**.

When you run the SA Installation script, the Installer prompts you to choose either the **Simple** or **Advanced** interview. If you choose Simple mode, the default values will be used for certain values, for example, passwords for the Oracle database, the Model Repository (`truth`) and Data Access Engine (`spin`) user, ports used by the Gateways, among others. In Advanced Mode, you can select values other than the default, giving you finer control.

Model Repository Prompts

The Model Repository is the database that stores information about the hardware and software deployed in the operational environment. Most of the Model Repository interview prompts apply only to a Single or First Core installation.

Table 28 lists the Model Repository prompts and the expected response.

Table 28 Model Repository Prompts

Prompt	Response
Please enter the service name (aka TNS name) of the Model Repository instance in the facility where the SA Installer is being run. **Parameter**: `truth.servicename`	Specify the service name, also known as the *alias*, for the Model Repository. For a Single Core, this is the server on which you are running the Installer. If you are installing the default Oracle database created by the Installer, the service name you provide here will be associated with the database during installation. If you intend to use an existing Oracle database, you can find the service name by looking in the `tnsnames.ora` file on the Model Repository instance. The service name is the value before the first equals sign (=) in the file. The location of this file can vary, so check with your DBA if you are not sure where to look. **Source**: The DBA who created the Oracle database. **Example**: `truth.example.com`

Table 28 Model Repository Prompts (cont'd)

Prompt	Response
Enter the service name (aka TNS name) of the Model Repository instance. **Parameter:** `slaveTruth.servicename`	Specify the service name, also known as the *alias*, for the core's Model Repository. You will see this prompt only when defining a new First Core. If this is a new installation, the service name you specify will be associated with the Model Repository during installation. If you plan to use an existing Model Repository, you can find the service name by looking in the `tnsnames.ora` file on the Model Repository instance. The location of this file can vary, so check with your DBA if you are not sure where to look. **Source**: The DBA who created the Oracle database. **Example**: `truth02.example.com`
Enter the SID of the Oracle instance that contains the Data Model Repository. **Parameter**: `truth.sid`	Specify the database system ID (SID) that was set when Oracle was installed on the server where the Model Repository is installed. If you are installing the HP-supplied Oracle database created by the Installer, the SID is `truth`. If you have an existing HP-supplied Oracle database, you will not be asked to supply this parameter. For an existing non-HP-supplied Oracle database, you can find the SID by looking in the `tnsnames.ora` file. The location of this file can vary, so check with your DBA if you are not sure where to look. **Source**: The DBA who created the Oracle database. **Example**: `DTC05`
Enter the path of the Oracle home directory. **Parameter**: `truth.orahome`	Specify the base directory of the Oracle database installation. If you are installing the HP-supplied Oracle database created by the Installer, the default location of `ORACLE_HOME` is `/u01/app/oracle/product/10.2.0/db_1`. If you have an existing HP-supplied Oracle database, you will not be prompted for this parameter. For an existing non-HP-supplied Oracle database, you can determine the Oracle home directory by logging in as the `oracle user` on the Model Repository server, and checking the value of the `$ORACLE_HOME` environment variable. (For a remote database installation, this parameter refers to the Oracle Client on the Model Repository server.) **Source**: The DBA who created the Oracle database. **Example**: `/u01/oracle/product/9.1 or /u01/app/oracle/product/10.2.0/db_1`

Table 28 Model Repository Prompts (cont'd)

Prompt	Response
Enter the fully-qualified path to the TNS admin directory (where the `tnsnames.ora` file resides). **Parameter**: `truth.tnsdir`	Specify the directory that contains the `tnsnames.ora` file. **Note**: This directory and path must be the same on all servers in a core. For example, since the Data Access Engine must access the `tnsnames.ora` file to connect to the Model Repository, the location of `tnsnames.ora` directory on the Data Access Engine server must be the same as the directory location on the Model Repository server. If you are installing the HP-supplied Oracle database created by the Installer, the `tnsnames.ora` file will be installed under `/var/opt/oracle.` If you have an existing HP-supplied Oracle database installed, you will not be prompted for this parameter. If you have an existing non-HP-supplied Oracle database, the location of the `tnsnames.ora` file can vary, so check with your DBA if you are not sure where to look. **Source**: The DBA who created the Oracle database. **Example**: `/var/opt/oracle`
Enter the fully qualified path to the directory where the export file will be saved. **Parameter**: `truth.dest`	*You must create this directory on the Model Repository server before you run the Installer.* Specify the directory in which the database export file will be saved. This directory must reside on the Model Repository server in the source facility. You will see this prompt only when installing a new First Core. **Note**: When adding a facility to a Multimaster Mesh, you must export the Model Repository from the source facility, then copy it to the destination facility. **Source**: Variable **Example**: `/export/home/core1`
Enter the fully qualified path to the directory that contains the export file. **Parameter**: `truth.sourcePath`	*This parameter is used when a new facility is added to a Multimaster Mesh and the source export file is copied to the new facility. This directory must exist on the server and contain the database export file before you run the Installer on the server.* Specify the directory on the destination facility's Model Repository server to which you copied the export data file from the source facility. **Source**: Variable **Example**: `/export/home/core2`

Table 28 Model Repository Prompts (cont'd)

Prompt	Response
Enter the IP address of the host where you want to install the Model Repository in the new facility. **Parameter**: `slaveTruth.truthIP`	Specify the IP address of the host on which you will install the Model Repository for the new target core. **Source**: Variable **Example**: `192.168.165.242`
Enter the IP address of the host where you want to install the Multimaster Infrastructure Components (vault). **Parameter**: `slaveTruth.vaultIP`	Specify the IP address of the host on which you will install the Model Repository Multimaster Component. The Model Repository Multimaster Component propagates and synchronizes changes from each Model Repository database to all other Model Repository databases **Source**: Variable **Example**: `192.168.165.242`

Database (Model Repository) Password Prompts

To ensure a secure installation of SA, the Installer prompts you to set passwords for numerous Oracle user accounts that the Core Components use to interact with one another. The passwords must meet the following standard Oracle criteria:

- The password cannot contain an Oracle reserved word (see Oracle's documentation for a full list).
- The password must be between 1 and 30 characters long.
- The password must start with a letter and use only alphanumeric and underscore (_) characters.

Table 29 lists the Database prompts and the expected responses.

Table 29 Database Password Prompts

Prompt	Response
Please enter the database password for the `opsware_admin` user. This password is used to connect to the Oracle database. **Parameter:** `truth.oaPwd`	Specify the `opsware_admin` password created by your database administrator. `opsware_admin` is an Oracle user that the Installer uses during installation to perform required tasks. If you are installing the HP-supplied Oracle database created by the Installer, the password you provide here will be associated with `opsware_admin` during installation of the database. If you have an existing Oracle database installation, this must be the password that your DBA set for the `opsware_admin` user when setting up the Oracle instance on the server. **Source**: Oracle DBA

Table 29 Database Password Prompts (cont'd)

Prompt	Response
Advanced Please enter the database password for the `lcrep` user. **Parameter**: `truth.lcrepPwd`	Specify the password for the `lcrep` database user. The Installer automatically creates an Oracle user `lcrep`, which SA uses internally for running multimaster replication between cores. The password you specify here will be associated with the `lcrep` user during installation. If you have an existing Oracle database installation, this must be the password that your DBA set for the `lcrep` user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`
Please enter the database password for the `gcadmin` user. **Parameter**: `truth.gcPwd`	Specify the password for the `gcadmin` database user. The Installer automatically creates an Oracle user `gcadmin`, which SA uses internally for removing old data from certain tables (referred to as the garbage collection process). If you have an existing Oracle database installation, this must be the password that your DBA set for the `gcadmin` user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`
Advanced Please enter the database password for the `truth` user. **Parameter**: `truth.truthPwd`	Specify the password for the Model Repository (`truth`) schema owner. The `truth` user is the main schema owner for the Model Repository and is automatically created by the Installer. If you have an existing Oracle database installation, this must be the password that your DBA set for the `truth` user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`

Table 29 Database Password Prompts (cont'd)

Prompt	Response
Advanced Please enter the database password for the `spin` user. **Parameter:** `truth.spinPwd`	Specify the password for the Data Access Engine (`spin`) user. **Note**: Passwords for the Data Access Engine (`spin`) user must be the same for all the cores in the mesh. The Installer automatically creates this database user. If you have an existing Oracle database installation, this must be the password that your DBA set for the Data Access Engine (`spin`) user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords **Example**: `x145_pwd03`
Advanced Please enter the database password for the `twist` user. **Parameter**: `truth.twistPwd`	Specify the password for the Web Services Data Access Engine (`twist`) user. The Installer automatically creates this user. If you have an existing Oracle database installation, this must be the password that your DBA set for the Web Services Data Access Engine (`twist`) user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`
Please enter the database password for the `vault` user. **Parameter:** `truth.vaultPwd`	Specify the Model Repository Multimaster Component (`vault`) user password. The Installer automatically creates the Model Repository Multimaster Component (`vault`) user. The Model Repository Multimaster Component propagates and synchronizes changes from each Model Repository database to all other Model Repository databases. If you have an existing Oracle database installation, this must be the password that your DBA set for the Model Repository Multimaster Component (`vault`) user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`

Table 29 Database Password Prompts (cont'd)

Prompt	Response
Advanced Please enter the database password for the `public views` user. **Parameter**: `truth.pubViewsPwd`	Specify the password for the `public_views` user, which SA uses for the Data Center Intelligence (DCI) module (server reporting). The DCI module uses this password when connecting with the Model Repository. The Installer automatically creates the `public_views` user. If you are using Brio™, Crystal Reports™, or other data reporting tools with the DCI module, you will be asked for the database user password when logging in to those applications so that you have read-only access to the Model Repository data. If you have an existing Oracle database installation, this must be the password that your DBA set for the `public_views` user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`
Advanced Please enter the database password for the `AAA` user. **Parameter**: `truth.aaaPwd`	Specify the password for the `AAA` user, (Access, Authentication, and Authorization (AAA) feature). The Installer automatically creates the `AAA` user. If you have an existing Oracle database installation, this must be the password that your DBA set for the `AAA` user when setting up the Oracle instance on the server. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`
Advanced Please enter the password to use for DCML Exchange Tool user. **Parameter:** `truth.detuserpwd`	Specify the password for the DCML Exchange Tool user (`DETUSER`). The Installer automatically creates the `DETUSER`. If you have an existing SA installation, this must be the password previously set for the `DETUSER`. **Source**: Variable, however, it must meet the requirements for Oracle passwords. **Example**: `x145_pwd03`

SA Component Password Prompts

Table 30 lists the password prompts for components other than the Model Repository and the expected responses.

If this installation is for a Multimaster Mesh, the following passwords must be the same for all cores belonging to the mesh.

Table 30 Component User and Password Prompts

Prompt	Response
Advanced Please enter the password for the Build Manager user. **Parameter:** `twist.buildmgr.passwd`	Specify the password for the Build Manager user (`buildmgr`). The `buildmgr` process will use this password when connecting to and authenticating with the Web Services Data Access Engine. The Installer automatically creates the Build Manager user (`buildmgr`). If you have an existing SA installation, this must be the password previously specified for the `buildmgr` user for the other cores in the mesh. **Password Restrictions**: The password cannot contain spaces or a forward slash (/). **Source**: Variable **Example**: `x145_pwd03`
Advanced Please enter the password for the `integration` user. **Parameter:** `twist.integration.passwd`	Specify the password for the `integration` user. Customers can use the `integration` user to access the SOAP APIs on the Web Services Data Access Engine. The Installer automatically creates the `integration` user. If you have an existing SA installation, this must be the password previously set for the `integration` user **Password Restrictions**: The password cannot contain a forward slash (/). **Source**: Variable **Example**: `x145_pwd03`

Table 30 Component User and Password Prompts (cont'd)

Prompt	Response
Please enter the password for the cryptographic material. **Parameter:** `decrypt_passwd`	Specify the password to use for decrypting cryptographic material. This password must be the same across all cores in a Multimaster Mesh. If you have an existing SA installation, this must be the password previously set for decrypting cryptographic material. **Password Restrictions**: The password cannot contain spaces and it must be between 4 and 20 characters long. **Source**: Variable **Example**: `x145_pwd03`
Please enter the password for the SA `admin` user. this is the password that will be used to authenticate the user `admin` to SA. **Parameter:** `cast.admin_pwd`	Specify the password for the SA `admin` user. **Password Restrictions**: This password cannot contain spaces. The Installer automatically creates the `admin` user. The first time you log in to the SAS Web Client to access a new Facility, you must log in as the `admin` user. **Source**: Variable **Example**: `x145_pwd03`

Facility Prompts

A *Facility* is a system object that represents a specific geographical location (such as Sunnyvale, Plano, Sacramento, or a data center). Servers and users are often associated with a facility as a means to enforce access rights and privileges. If you are performing a Single Core installation, your deployment is a single facility. Multimaster installations, however, consist of two or more facilities.

In this section, the first core installed in a Multimaster Mesh is called the *Primary Core*, and is the core that has the first Model Repository installed. *Secondary Cores* are the second, third, and fourth (and so on) cores installed in the mesh. For historical reasons, Primary Cores are sometimes referred to in parameter names as *Master* and Secondary Cores as *Slave*.

Table 31 lists the Facility prompts and the expected responses.

Table 31 Facility Prompts

Prompt	Response
Please enter the authorization domain. **Parameter:** `truth.authDom`	Specify the authorization domain for the initial (default) customer. This value is usually the same as the domain name. The domain name must be uppercase, less than 50 characters, and in domain name format. **Important**: You must use the same value for every core in your Multimaster Mesh. The Installer prompts you for this value only when you are installing your first SA Core. If you convert to a Multimaster core, the authorization domain will be picked up from the response file and carried over to all subsequent core installations. **Source**: Variable **Example**: `XYZ.COM`
Please enter the subdomain for this facility (lowercase, no spaces). **Parameter:** `truth.dcSubDom`	Specify the fully-qualified DNS subdomain where the core is to be deployed. This is the facility where you run the Installer. This value must be unique for each core in the Multimaster Mesh. The value is based on the VLAN for the facility in which you are installing the core. The subdomain name must be in lowercase, less than 50 characters long, and in subdomain format. **Source**: Your network administrator. **Example**: `dc1.example.com`
Enter the subdomain for the facility you are about to create (lowercase, no spaces). **Parameter:** `slaveTruth.dcSubDom`	Specify the fully-qualified DNS subdomain where the Destination Multimaster Core is to be deployed. This value must be *unique* for each core in the Multimaster Mesh, both Source and Destination Cores. The value is based on the VLAN for the facility in which you are installing the Multimaster core. The subdomain name must be in lowercase, less than 50 characters, and in subdomain format. **Source**: Your network administrator. **Example**: `dc2.example.com`

Table 31 Facility Prompts (cont'd)

Prompt	Response
Enter the short name of the facility where the SA Installer is being run (no spaces). **Parameter:** `truth.dcNm`	Specify the short name of the facility where the Installer is being run. This would also be the location of the Primary Core. Some SA processes use this name internally. It must be in uppercase, less than 25 characters, and cannot contain spaces or special characters (underscores are allowed, dashes are *not* allowed). **Source**: Variable **Example**: HEADQUARTERS
Multimaster Enter the short name of the new facility you would like to define **Parameter**: `slaveTruth.dcNm`	Specify the default facility name for the Secondary Core. Some SA processes use this name internally. It must be less than 25 characters, and cannot contain spaces or special characters (both dashes and underscores are allowed). **Source**: Variable **Example**: NORTHSIDE
Please enter the default locale for users of the Command Center (en/ja) **Parameter**: `default_locale`	Specify the default locale for the SAS Web Client. For example, the locale entry `en` sets the language, character set, and date-and-time formats to English. **Source**: In this release, the allowed values are `en` (English) and `ja` (Japanese). **Example**: `en` or `ja`
Advanced Please enter the facility long name. **Parameter**: `truth.dcDispNm`	Specify the name that will display in the SAS Web Client title. This is the facility where Primary Core is located. **Name Restrictions**: The name must be unique, less than 50 characters, and cannot include any special characters (`< > () & * \ ' ?`). **Source**: Variable **Example**: Los Angeles Office
Advanced Enter the long name for the facility that you are adding to the mesh. **Parameter:** `slaveTruth.dcDispNm`	Specify the name of the Secondary Core that displays in the SAS Web Client title. **Name Restrictions**: The name must be unique, less than 50 characters, and cannot include any special characters (`< > () & * \ ' ?`). **Source**: Variable **Example**: Toronto Office

Table 31 Facility Prompts (cont'd)

Prompt	Response
Please enter the Facility ID (number only, less than or equal to 999, with no leading zeros). **Parameter**: `truth.dcId`	Specify an ID that uniquely identifies the facility. When you install the Primary Core, you will be prompted to provide this ID. When you install subsequent Secondary Cores in the same Multimaster Mesh, SA automatically generates the Facility ID when you add a new facility using the SAS Web Client. You can determine the Secondary Core's Facility ID by logging in to the SAS Web Client at the Primary Core facility, then select **Opsware Facilities** under **Environment** in the Navigation pane and click the facility's name. **ID Restrictions**: The Facility ID value is capped at 1000. Therefore, you must specify a number for the first facility that is far enough below 1000 that you will have sufficient IDs available to continue adding facilities to your Multimaster Mesh. **Source:** Variable for the first facility; set by the SA for subsequent facilities. **Default**: 1

SA Feature Prompts

The responses to the following prompts will be used to configure the SA features: OS Provisioning, Software Provisioning, Patch Management, and NAS Integration.

Table 32 lists the SA Feature prompts and the expected responses.

Table 32 SA Feature Prompts

Prompt	Response
Please enter the directory that contains the Microsoft utilities. (Press Ctrl-I for a list of required files) **Parameter**: `windows_util_loc`	Specify the directory to which you have already copied the Microsoft utilities required for Window's Patch Management (`qchain.exe`, `mbsacli20.exe`, `wusscan.dll`, `WindowsUpdateAgent20-x86.exe` and `wsusscan.cab` files). **Note**: If you have not yet copied these Microsoft utilities onto the server you will use for OS Provisioning, see Windows Patch Management Requirements on page 62 **Source**: Variable, however, this directory *must* exist on the same server as the Software Repository (part of the Slice Component bundle). **Example**: `/home/win_util`

Table 32 SA Feature Prompts (cont'd)

Prompt	Response
Please enter the OS Provisioning Boot Server IP address or hostname. **Parameter**: `bootagent.host`	Specify the server on which you installed the OS Provisioning Boot Server. Important: You must provide a valid IP address or host name that can be resolved from the server on which you installed the OS Provisioning Boot Server component and the Build Manager. Additionally, the host name must be resolvable by SA managed servers for OS provisioning. Source: Variable Example: `foo.example.com`
Enter the default network speed/duplex setting for Solaris servers. **Parameter:** `boot_server.speed_duplex`	Specify the default network speed and duplex that should be used by Solaris servers booted from the OS Provisioning Boot Server. **Valid Responses**: `100fdx`, `100hdx`, `10fdx`, `10hdx`, `100T4`, and `autoneg`. Enter a value without spaces. **Source**: Variable **Example:** `100fdx`
Please enter the pathname to the Linux media. **Parameter**: `media_server.linux_media`	Specify the path to the Linux OS media on the server on which the Media Server will be installed. Providing the path to the Linux OS media does not actually copy the media to the Media Server. See the *SA Policy Setter Guide* for the steps required to set up the media on the Media Server for OS Provisioning. Source: Variable, however, this directory must exist on the server where the Media Server is installed. Example: `/home/os_media/linux/`
Please enter the pathname to the Solaris OS media. **Parameter**: `media_server.sunos_media`	Specify the path to the Sun Solaris OS media on the server on which the Media Server will be installed. Providing the path to the Solaris OS media does not actually copy the media to the Media Server See the *SA Policy Setter Guide* for the steps required to set up the media on the Media Server for OS Provisioning. Source: Variable, however, this directory must exist on the server where the Media Server is installed. Example: `/home/os_media/solaris/`

Table 32 SA Feature Prompts (cont'd)

Prompt	Response
Please enter the pathname to the Windows OS media. **Parameter**: `media_server.windows_media`	Specify the path to the Microsoft Windows OS media on the server on which the Media Server will be installed. The OS Provisioning feature exports Windows OS media to SMB clients through a Samba share. Providing the path to the Windows OS media does not actually copy the media to the Media Server. See the *SA Policy Setter Guide* for the steps required to set up the media on the Media Server for OS Provisioning. Source: Variable, however, this directory must exist on the server where the Media Server is installed. Example: `/home/os_media/windows/`
Advanced Please enter the share name to use for the Windows Media Sharing Server. **Parameter**: `media_server.windows_share_name`	Specify the share name that Samba to will use to export the Windows OS media. **Name Restrictions:** Share names that are longer than eight (8) characters can give errors while browsing or may not be accessible to some older clients. The share name is not case sensitive. **Source**: Variable **Example**: `WINMEDIA`
Advanced Please enter a password to write-protect the Windows media share. The Import_media tool will prompt for this password each time it is run. **Parameter**: `media_server.windows_share_password`	Specify the `root` user password, which enables write access to the Windows share. The Import Media Tool prompts for this password each time it is run. **Password Restrictions**: The password cannot contain spaces. **Source**: Variable **Example**: `x145_pwd03`
Please enter the root directory for the Package Repository. **Parameter**: `word_root`	Specify the directory in which to store Software Provisioning packages on the Software Repository (Slice Component bundle) host. **Note**: Ensure that this directory has sufficient free disk space. **Source**: Variable **Example**: `/var/opt/opsware/word`
Please enter the host name or IP address of the Network Automation (NA) server. (Enter "none" if NA is not installed.) **Parameter**: `twist.nasdata.host`	Specify the host name or IP address of the server running HP Network Automation (NA), if installed. If NA is not installed, accept the default value none. Enter a value without spaces. **Source**: The network administrator/SA administrator who installed HP Network Automation. **Example**: `192.168.165.242`

Table 32 SA Feature Prompts (cont'd)

Prompt	Response
Please enter the location where the Software Repository temporarily places content during uploads. **Parameter** `word_tmp_dir`	Specify the fully qualified location for the temporary storage of content during Software Repository uploads. **Source**: Variable **Example**: `/var/tmp/uploads`
Please enter the host (NFS server) where Software Repository Content resides. **Parameter** `word.store.host`	Specify the host name of the server where Software Repository content is stored. **Source**: Variable **Example**: `192.168.165.243`
Please enter the path to the server where Software Respiratory content resides. **Parameter** `word.store.path`	Specify the path to the server where Software repository content is stored. This will be to the server specified in word.store.host. **Source**: Variable

SA Gateway Prompts

The responses to the following prompts will be used to configure the IP addresses and ports at which SA Gateways can be contacted by Core Components, Agents, or other gateways.

Table 33 lists the gateway prompts and valid responses.

You can use only port numbers below 64001.

Table 33 SA Gateway Prompts

Prompt	Response
Please enter the IP address of the Management Gateway. **Parameter**: `mgw_address`	Specify the IP address of the Management Gateway. The Management Gateway manages Core-to-Core communications. Core Gateways installed on Secondary Cores and/or Satellite Gateways also communicate with the Management Gateway. **Source**: Variable **Example**: `192.168.165.242`
Advanced Please enter the port on which the Core Gateway will listen for connections from other Gateways. **Parameter**: `cgw_slice_tunnel_listener_port`	Specify the port number on which the Slice Component Core Gateway listens for connections from other Gateways. **Source**: Variable **Default**: 2001

Table 33 SA Gateway Prompts (cont'd)

Prompt	Response
Advanced Enter the port on which the Management Gateway will listen for connections from other gateways. **Parameter**: `mgw_tunnel_listener_port`	Specify the port on which the Primary and Secondary Cores' Management Gateways will listen for connections from other Core and Satellite gateways. **Source**: Variable **Example**: 2001
Please enter the port on which the Management Gateway in the Primary Core listens for connections from other Gateways. *Parameter:* `masterCore.mgw_tunnel_listener_port`	Specify the port number on which the Primary Core's Management Gateway listens for connections from other Gateways. This port will be used during installation of Secondary Cores to create a Multimaster connection between the Management Gateways on the Primary and Secondary Cores. **Source**: Variable **Example**: 2001
Advanced Enter the port on which the Management Gateway can be contacted to request connections to Core Components. *Parameter:* `mgw_proxy_port`	Specify the port number through which Core Components can request tunneled connections to other components through the Management Gateway. Source: Variable Example: 3003
Advanced Please enter the port for the administrative interface for the Core Gateway. *Parameter:* `cgw_admin_port`	Specify the communication port for the Core Gateway's administrative interface or accept the default. Source: Variable Default: 8085 Example: 8085
Please enter the port on which Server Agents can contact the Agent Gateway to request connections to Core Components. **Parameter**: `agw_proxy_port`	Specify the port number through which Server Agents request connections from the Agent Gateway to Core Components. **Source**: Variable **Example**: 3001
Please enter the port on which Core Components can contact this Core Gateway to request tunneled connections. **Parameter**: `cgw_proxy_port`	Specify the port number through which core components can request tunneled connections from the Core Gateway. **Source**: Variable **Example**: 3002

Global File System Prompts

The responses to the following prompts will be used to configure IP addresses and directories for the Global File System.

Table 34 lists the Global File System prompts and the expected responses.

Table 34 Global File System Prompts

Prompt	Response
Advanced Please enter the IP or host name of the NFS server for the `Global File System` user (user, home, and temp directories). **Parameter**: `ogfs.store.host.ip`	Specify the IP address or host name of the NFS server from which Global File System `/usr, /home` and `/tmp` directories are to be mounted. **Source**: Variable **Example**: `192.168.198.92`
Advanced Please enter the absolute path on the NFS server for the `Global File System` user (user, home, and temp directories). **Parameter**: `ogfs.store.path`	Specify the absolute path to the `/usr, /home` and `/tmp` directories for the Global File System. As of SA 7.80, the value for `ogfs.store.path` cannot be the same as `ogfs.audit.path` or the upgrade/install will fail. **Source**: Variable **Example**: `/var/opt/opsware/ogfs/export/store`
Advanced Please enter the IP or host name of the NFS server for the Global File System where the audit streams will be stored. **Parameter**: `ogfs.audit.host.ip`	Specifies the IP address of the server where storage for audit streams for the Global File System will be mounted. **Source**: Variable **Example**: `192.168.165.242`
Advanced Please enter the absolute path on the NFS server for the Global File System where the audit streams will be stored. **Parameter**: `ogfs.audit.path`	Specify the absolute path for the storage of the audit streams for the Global File System. As of SA 7.80, the value for `ogfs.store.path` cannot be the same as `ogfs.audit.path` or the upgrade/install will fail. **Source**: Variable **Example**: `/var/opt/opsware/ogfs/export/audit`
Please enter the pathname of where you wish the local cache of snapshots and audits to be. This will require a large amount of disk space (4 Gb by default). **Parameter**: `spoke.cachedir`	Specify the directory win which the Global File System service will cache snapshots and audits for quick access. **Default:** `/var/opt/opsware/compliancecache` **Source**: Variable **Example**: `/var/opt/opsware /compliancecache`

Table 34 Global File System Prompts (cont'd)

Prompt	Response
Advanced Please enter the *minimum ID number* to use when assigning Unix user IDs to SA users. **Parameter:** `twist.min_uid`	Specify the minimum UID number that can be used. Unix UIDs are automatically generated for each SA user. UIDs will be allocated by counting up from the minimum UID. **Valid Values**: — Numeric only — Minimum — 1024 — Maximum — 90000000 — No leading zeroes **Default**: 80001 **Source**: Variable **Example**: `80001`
Advanced Please enter the *default Unix group* ID to assign to SA users. **Parameter:** `twist.default_gid`	Specify the default group ID number that is assigned when an SA user is created. To restrict SA users from using certain ports, this group ID has the least amount of network privileges. The default value is 70001. **Valid Values**: — Numeric only — Minimum — 1024 — Maximum — 90000000 — No leading zeroes **Default**: 70001 **Source**: Variable **Example**: `70001`

Uninstallation Prompts

Table 35 lists the prompts lists the Database prompts and the expected responses for an uninstallation of an SA core.

Table 35 Uninstallation Prompts

Prompt	Description
Do you need to preserve any of the data in this database? **Parameter**: `truth.uninstall.needdata`	Uninstalling the Model Repository permanently deletes all data in the database, therefore, the uninstallation process stops if you reply "Yes" to this prompt. If you want to do an uninstallation, backup your data, run the uninstallation again and answer "No" to this prompt. Remember, the Installer *does not* preserve any data. **Example**: `y`

Table 35 Uninstallation Prompts (cont'd)

Prompt	Description
Are you sure you want to remove all data and schema from this database? **Parameter:** `truth.uninstall.aresure`	Uninstalling the Model Repository permanently deletes all data in the database, you can stop the uninstallation by responding "no" to this prompt.
Would you like to preserve the database of cryptographic material? **Parameter**: `save_crypto`	If you answer yes, the database of cryptographic material is saved. If you answer no, the material is deleted as part of the uninstallation. Example: y
Are you absolutely sure you want to remove all packages in the repository? **Parameter:** `word.remove_files`	If you answer yes, the packages, logs, and cryptographic material for the Software Repository are removed. **Example**: `y`

Using the SA Installer

This section discusses the following topics:

SA Installation Media

SA is available on and installable from the following DVD set that contains the scripts for installing, uninstalling, and upgrading components.

- **Product Software DVD**: Contains all packages and scripts necessary to install an SA core, including the HP-supplied Oracle RDBMS.
- **Agent and Utilities DVD**: Contains the Agents and utilities (such as the OS Provisioning Boot Agent, SA Agents for various operating systems, and so on) that must be uploaded to the Software Repository after the SA core has been installed.
- **Satellite Base DVD**: Contains the packages and scripts required to install a Satellite Core including an SA Gateway and a Software Repository Cache.
- **Satellite Base Including OS Provisioning DVD**: Contains the packages and scripts required to install a Satellite Core including an SA Gateway, a Software Repository Caches, as well as the OS Provisioning components.

For reference, the script names are listed in the section, Installer Command Line Syntax on page 97.

The Product Software DVD and the Agent and Utilities DVD require a Dual Layer DVD drive.

Copying the DVDs to a Local Disk

It is recommended that you copy the contents of the SA DVDs to a local disk or to a network share and run the Installer from that location.

When you copy the contents of a SA DVD to a local disk or the network, you must create a directory structure that duplicates the structure of the DVD, for example:

```
/opsware_system
```

The path of the directory cannot have spaces.

Although you run the Installer from the common parent directory, `/opsware_system`, the Installer will change to other directories as needed during the installation.

Installer Command Line Syntax

You invoke the Installer using one of the following three scripts:

- `install_opsware.sh` — installs the Oracle database and Model Repository, installs the Core Components for a First Core, installs the components for subsequent cores, exports the contents of the Model Repository.
- `upgrade_opsware.sh` — upgrades a Core Component(s) to a new version.
- `uninstall_opsware.sh` — uninstalls a single Core Component or uninstalls all Core components.

All three of these scripts run with the same command line arguments, as Table 36 shows:

Table 36 *SA* Installer Command Line Arguments

Argument	Description
`-h`	Display the Installer help for the command line options. *To display help during the interview, press `ctrl-I`.*
`--resp_file=`*`file`* `(-r` *`file`*`)`	Invoke the Installer using the values in the specified response file. You will create and save the response file for an installation the first time you run the installer. The installer prompts for the component to install and then runs an interview that only prompts for data missing from the specified the response file. If the response file is incomplete, the installer prompts for the missing information. The installer keeps an inventory of the components that are installed on a given server.

Table 36 *SA* Installer Command Line Arguments (cont'd)

Argument	Description
`--interview`	Conduct the installation in interview mode. You will be prompted to provide values for a number of component parameters. At the end of the interview, the installer saves the response file. Typically, you specify this option when you run the Installer for the first time. You can also specify this option when you have an incomplete response file. If you specify both the `--interview` and `--resp_file` options, the installer runs the interview but uses the values in the response file you specified as the defaults. `--interview` is the default.
`--verbose`	Run the installer in verbose mode which causes more information to be displayed on the console. See also Installer Logs on page 99.

Installer Interview Modes

When you run the Installer in interview mode, you will be prompted to choose the Simple or Advanced interview.

Simple Interview

if you choose the Simple Interview, the default values for certain parameters that are rarely modified will be used (you will not be prompted to specify values for these parameters). These parameters include the various Oracle passwords used internally by the Core Components.

Advanced Interview

If you choose the Advanced Interview, the installer prompts you to supply values for *all* parameters that are relevant to the type of installation.

The Interview Process

The installer validates certain responses to the interview prompts as you enter them; you will be asked to re-enter a value if the installer is not able to validate your response (for example, a directory or path that does not exist or an invalid value or range). Some parameters are also revalidated during the actual installation of the Core Components. If a response to a prompt cannot be validated at time of installation, the installer runs a mini-interview during which you can provide a valid response.

Help

At any time during the interview, you can press `ctrl-I` to display help for the current interview prompt. A brief description of the prompt and the expected responses will be displayed.

Concluding the Interview

After you have responded to all the prompts and have provided values for all parameters, the installer asks if you want to finish the interview.

You can go back to review or change your answers by pressing “n”. If you press “y”, the installer prompts you top provide the fully qualified file name and path name for the response file in which it will save your answers. Ensure that the directory in which the response file is to be saved exists.)

Continuing an Installation without Exiting the Installer

After you save the response file, the installer asks if you would like to continue the installation using the data from the response file you just saved. If you press “y”, the installer displays Component Installation screen. If you press “n”, the installer exits.

Reusing a Response File

When you start the Installer, you can specify the response file to use during the installation by invoking the installer using the `--resp_file=`*`file`* (or `-r file`) parameter and specifying the fully qualified path to the response file. The installer will read the response file and use the parameter values stored in that file during the installation.

When you install a core on multiple servers, you should copy the response file from the First Core installation to the other servers so that the installations of subsequent components can use the same data from that response file. This is useful because many parameter values, and directory, file, and path names must be identical on all servers in the Core.

Installer Logs

The Installer logs output to the console on which it is run and to a standard log file:

`/var/log/opsware/install_opsware/install_opsware.`*`timestamp`*`.log`

By default, it also generates a more verbose version to:

`/var/log/opsware/install_opsware/install_opsware.`*`timestamp`*`_verbose.log`

If you specify the `--verbose` option, the output to the console will be more verbose while the contents of the standard and verbose log files will remain the same.

Some Core Components have supplementary logs that contain additional details about the installation of those components.

See the *SA Administration Guide* for information about the logs for Core Components.

The following log files are created during the installation of the Model Repository:

`/var/log/opsware/install_opsware/truth/truth_install_`*`number`*`.log`
`/var/log/opsware/install_opsware/truth/truth_install_`*`number`*`_verbose.log`

When you install a First Core, it is recommended as best practice that you open a second terminal window and issue the following command:

`tail -f /var/log/opsware/install_opsware/install_opsware.<date>_verbose.log`

Where `<date>` is the most recent timestamp.

Obfuscating Cleartext Passwords

During the SA installation or upgrade process, some cleartext passwords will be automatically obfuscated and some will not. Some passwords will be obfuscated when SA Core Components start up, such as the OS Provisioning Build Manager password when the Web Services Data Access Engine server starts up. Some passwords in certain files will not be obfuscated, such as passwords in the installation logs and Installer response files.

There are several ways to manually secure cleartext passwords. Which you choose will depend on your security requirements:

- Encrypt the response files and installation logs.
- Purge sensitive information from the Installer response files.
- Store the Installer response files and logs on a secure server.

Table 37 lists cleartext passwords that are automatically obfuscated and passwords that must be manually secured.

Table 37 Cleartext Passwords

Cleartext Password	Filename	Automatically Obfuscated	Manually Secured
admin	`/var/opt/opsware/twist/?DefaultAuthenticatorInit.ldift`	✔	

buildmgr	`/var/opt/opsware/crypto/buildmgr/twist.passwd`
	`/var/opt/opsware/crypto/occ/twist.passwd`
	`/var/opt/opsware/twist/?DefaultAuthenticatorInit.ldift`
		✔
		✔
		✔	

cleartext admin	`/etc/opt/opsware/twist/startup.properties`	✔	
detuser	`/var/opt/opsware/crypto/twist/detuserpwd` `/var/opt/opsware/crypto/OPSWhub/twist.pwd`	✔ ✔	
integration	`/var/opt/opsware/twist/?DefaultAuthenticatorInit.ldift`	✔	
root	`/var/log/opsware/agent/agent.err`		✔

Table 37 Cleartext Passwords (cont'd)

Cleartext Password	Filename	Automatically Obfuscated	Manually Secured

	Installer response files:
	`/var/opt/opsware/install_opsware/resp`
	`/var/opt/opsware/install_opsware/install_opsware*`
	`/var/tmp/@*`
	`/var/opt/opsware/install_opsware/truth/truth_install_*`
		✔	✔
			✔
			✔

spin	`/etc/opt/opsware/spin/spin.args`	✔	
vault	`/var/opt/opsware/crypto/vault/vault.pwd`	✔	

Securing Installer Log and Response Files

Depending on the level of your security requirements, it is recommended that the installation or upgrade team encrypt or move installation logs files to a secure server and, if necessary, encrypt, move to a secure server, and/or purge sensitive information from the Installer response file. Remember that the response file will be needed for upgrades and subsequent Core installations and the log files are useful for troubleshooting so completely removing them is not recommended.

The Installer reminds you to protect sensitive log files by displaying the following message at the end of the installation process:

```
######################################################################
WARNING: to make sure that no sensitive information is left
on this server, please encrypt or copy to a secure location the following
files and directories:
  -- /var/opt/opsware/install_opsware/resp/*
  -- /var/log/opsware/install_opsware/*
  -- /var/tmp/*.sh
Also, please encrypt or store in a secure location the response file that you
used to install this core.
######################################################################
```

6 Installing the First Core

This section describes the installation tasks for a First Core (formerly standalone core). The topics covered include:

First Core Installation Basics

This section describes how to install the First Core for a Facility. This core can be

- A single core that manages servers in a single Facility
- The first (primary) core of a Multimaster Mesh installation

Whether you will be using a single core to manage servers in a single Facility or a Multimaster Mesh to manage servers in multiple facilities, you will need to perform the tasks described in this section to install the First Core.

- If you are planning a *single core* in a single Facility, after you complete the tasks in this section, your core will be up-and-running and you will be ready to manage servers in your Facility.
- If you plan this core to be the first in a *Multimaster Mesh installation*, after you complete the tasks in this section to install the First Core of the mesh, you will need to complete the tasks described in Chapter 8, “Multimaster Mesh Installation” to add additional cores to your mesh.

A First Core has all the components required to become the primary core of a Multimaster Mesh — there is no Multimaster conversion required as in earlier SA releases. You simply need to add subsequent cores and configure them to communicate with the First Core.

In a Multimaster Mesh installation, the First Core functionality is not much different than all the other cores in the mesh, however, it does have certain central components that oversee communication between the various cores as well as manage conflicts and load balancing.

A core’s component bundles can be installed on different servers for performance scalability but the core is still seen as a single logical entity. Certain components can also be unbundled and distributed among different hosts by using the SA Custom Installation.

Overview of the Installation Process

A typical First Core installation has the following phases:

1. *Pre-Installation*: Ensure that all installation pre-requisites have been met, that you have the information needed to complete the SA Installer interview, that you have all necessary permissions to complete the installation, and that you have the SA installation DVDs. For more information, see Chapter 3, "Pre-Installation Requirements" and Chapter 5, "Prerequisites for the Installer Interview".
2. *Database Installation*: The Model Repository requires that an Oracle database is available before the SA Installer is run. You can:
   - install the *HP-supplied Oracle database* provided with the SA product software
   - use an *existing Oracle database installation* that you have configured for use with SA
   - install the database using the *Oracle Universal Installer* before beginning the SA installation.

   (For details about installing the required database, see Appendix A, "Oracle Setup for the Model Repository").
3. *Installation Interview*: When you install your First Core, you will run the Installer script in Interview Mode. During this process, you be required to provide information about your environment. At the end of the process, the information will be saved in a Response File that will be used to complete the installation. This response file is also used to install Subsequent Cores, Satellites, and for upgrades.
4. *Core Component Installation*: You will run the Installer and select the SA components to install. In this step, the Installer creates the SA directories and files on a server. If your core is to be on a single server, you need only run the Installer once. If you plan to distribute your Core Components to multiple servers, you log on to each server and run the Installer to install the components for that server.
5. *Upload Software Repository Content*: On the server where you installed the Software Repository (part of the Slice Component bundle), you upload the default SA Software Repository content.
6. *Post-Installation*: You will complete the post-installation tasks.

   For more information, Chapter 7, "First Core Post-Installation Tasks".

Oracle Database Installation Options

A functioning, properly configured Oracle 10g or 11g database must be available *before* you begin the SA installation process. You can choose to:

- Use the HP-supplied Oracle 11g database and allow the SA Installer to install and pre-configure the database. For more information, see Appendix A, "Oracle Setup for the Model Repository".

As of SA 7.50, the Oracle database is distributed on its own DVD.

- Use the Oracle Universal Installer to install an Oracle 10g or 11g database. For more information, see Appendix A, "Oracle Setup for the Model Repository".
- Use an existing Oracle 10g or 11g installation. This database must be configured for use with the SA Model Repository. For more information about the required configuration, see Appendix A, "Oracle Setup for the Model Repository". You may need to contact your local Oracle DBA for assistance with integrating SA with your pre-existing Oracle database.

If you choose to install the HP-supplied Oracle database, the *Oracle_SAS* Installer will guide you through the process.

If you choose use the Oracle Universal Installer to install Oracle, you must do so before running the SA Installer, making sure to record all database-related information required by the Installer Interview, such as passwords, the path to `ORACLE_HOME`, and so on.

The version of the HP-supplied Oracle database is Oracle Database Standard Edition 11g. For manual installations, SA supports both the Oracle Database Standard Edition and the Oracle Database Enterprise Edition.

SA Component Bundles

Certain SA Core Components are *bundled* together and installed as a *unit*. You can, however, install multiple instances of the Slice Component bundle for scalability. You can also use the SA Custom installation to unbundle and distribute certain Core Components to different servers. See SA Core Component Bundling on page 15 for more information about component bundles.

Table 38 shows how components are bundled in SA Slices

Table 38 SA Bundles and Core Component Distribution

Model Repository	Infrastructure Components	OS Provisioning Components	Slice Components Bundle #1	Slice Component Bundle #2
One per core	One per core	Typically one per core	Multiple per core	Multiple per core
Model Repository	Management Gateway, Primary	Media Server	Core Gateway/Agent Gateway	Core Gateway/Agent Gateway
	Data Access Engine	Boot Server	Command Center	Command Center
	Model Repository Multimaster Component		Global File System	Global File System
	Software Repository Store (*can be optionally located on another host*)		Web Services Data Access Engine	Web Services Data Access Engine
			Secondary Data Access Engine	Secondary Data Access Engine
			Build Manager	Build Manager
			Command Engine	Command Engine
			Software Repository	Software Repository

Installation Tips

Before you invoke the SA Installer, you should have:

- Planned your SA Core Component deployment. When planning a core deployment, decide whether you want to install the Core Components on a single server or distribute them to multiple servers or whether you will need multiple instances of the Slice Component bundle. See Chapter 1, "SA Architecture" and SA Core Performance Scalability on page 41.
- Performed the pre-installation administration tasks, such as configuring your network and verifying operating system, package/utility, and hardware and software availability/compatibility. See Chapter 3, "Pre-Installation Requirements."
- Gathered the information necessary to complete the Installer interview. See Chapter 5, "Prerequisites for the Installer Interview."
- Verified that the Model Repository host meets the prerequisites described in the following sections:
  - Supported Oracle Versions on page 188
  - Hardware Requirements on page 189
  - Operating System Requirements on page 191
- Installed the Oracle database.

First Core Installation Phases

This section provides a summary of the Fist Core upgrade process. You can use the right-hand column to indicate that a phase is completed:

Table 39

Phase	Description	Complete
1	Preparing to Install a First Core	
2	Invoke the SA Installer and Complete the SA Installer Interview	
3	Install the Core Components	
4	Post-Component Installation Tasks	
5	Upload the Software Repository Content	

First Core Installation Procedure

A VMware ESX/ESXi server guest OS (virtual machine) is *not* supported as a Core Server.

This section contains instructions for running the Installer to install a First Core. The installer is a script called `install_opsware.sh` and is located on your SA distribution media.

Complete the following tasks to install a First Core:

Phase 1: Preparing to Install a First Core

1. You will need the *SA Product Software DVD* and the *SA Agent and Utilities DVD*. If you plan to install the *HP-supplied Oracle database*, you will need the *Oracle_SA installation DVD*.

   See SA Installation Media on page 96, including the recommendation, "Copying the DVDs to a Local Disk."

2. On the server where you will install the new SA Core Components (or on each server that will host Core Components if you plan to distribute components to different hosts), mount the *Product Software* DVD and, optionally, the *Oracle_SA* DVD, or NFS-mount the directory that contains a copy of the DVD contents.

The SA Installer must have *read/write root access* to the directories where it will install the SA components, including NFS-mounted network appliances.

3. On the server where you will install the Model Repository, open a terminal window and log in as `root`.
4. Change to the root directory:

   ```
   cd /
   ```

Installing the HP-Supplied Oracle Database for the Model Repository

If you plan to use the HP-supplied Oracle database (for the Model repository), you must complete the tasks in Installing the HP-Supplied Oracle RDBMS Software and Database on page 195.

Using the Oracle Universal Installer or Manually Installing a Oracle Database

If you plan to use the Oracle Universal Installer to install your database or plan to use an existing Oracle 10g or 11g database, ensure that your database meets the requirements for use with the SA Model Repository described in Appendix A, "Oracle Setup for the Model Repository" and follow the installation instructions in Using the Oracle Universal Installer or Manually Installing a Oracle Database on page 107 and Manually Creating the Oracle Database on page 201.

OS Provisioning

If you plan to use OS Provisioning, before you run the SA Installer interview for the First Core, the paths for the OS provisioning media must already exist on the server where you will install the OS Provisioning components even if the content has not yet been uploaded.

Phase 2: Invoke the SA Installer and Complete the SA Installer Interview

1. Invoking the SA Installer for the first time depends on how you installed the Model Repository database:

   a. If you installed the *HP-supplied Oracle database*, you have the response file you created during that installation (default: `oiresponse.oracle_sas`). On the server that will be your First Core or, if you plan to distribute components on different servers, on the server that will host the *Model Repository*, start the installation with the following command:

   ```
   /opsware_system/opsware_installer/install_opsware.sh -r
   <response_file_name>
   ```

During the interview phase below, the database information in the response file is used to complete certain of the parameter default values.

You must specify the full path to the script. The example above assumes that you have copied the contents of the *SA Product Software DVD* to a local disk or network share.

b If you installed the Oracle database using the *Oracle Universal Installer* or are using an existing Oracle installation, *you will not have a response file*. You must supply all the Oracle-specific parameter values during the interview based on your Oracle configuration. On the server that will be your First Core or, if you plan to distribute components on different servers, on the server that will host the *Model Repository*, run the SA Installer in *Interview Mode* by invoking it with no command-line argument:

```
/opsware_system/opsware_installer/install_opsware.sh
```

You must specify the full path to the script. The example above assumes that you have copied the contents of the *SA Product Software DVD* to a local disk or network share.

2 The Installer Installation Options screen displays the following:

```
Welcome to the Opsware Installer. Please select one of the
  following installation options:

1 - Multimaster Opsware Core - First Core
2 - Multimaster Installation: Define New Facility; Export Model Repository
3 - Multimaster Opsware Core - Subsequent Core
```

When you install an SA Core, HP recommends that you open a second terminal window and issue the following commands:

```
# cd /var/log/opsware/install_opsware/install_opsware
# tail -f install_opsware.<date>_verbose.log
```

where `<date>` is the most recent timestamp. This will allow you to see all messages posted to the log by the installer as the installation progresses.

3 At the installation options prompt, select the option:

```
1 - Multimaster Opsware Core - First Core
```

Press `c` to continue.

Note that, to install a First Core you must select *Multimaster Opsware Core - First Core* even if you do not plan to install subsequent Secondary Cores. A First Core is a fully Multimaster-capable core, even if you use it as a standalone core and do not use the Multimaster functionality.

4 The Installer Component Layout Mode screen displays the following:

```
Please select the component layout mode. In a "typical" install,
components are already bundled together in a pre-defined configuration.
"Custom" install allows you to install components "a la carte."

1 - Typical Component Layout Mode
2 - Custom Component Layout Mode
```

5 At the Component Layout Mode Prompt, select the option:

```
1 - Typical Component Layout Mode
```

Choosing *Option 2, Custom Component Layout Mode*, gives you the ability to more finely control the distribution of the SA Core Components by breaking certain components out of their component bundles.

You should use this option only if you are very certain that you understand how to distribute Core Components across Core Servers. Be aware that breaking up the component bundles can make diagnosing and troubleshooting problems later more difficult.

Complete the Installation Interview

The following steps use a Typical Layout and a Simple Interview.

1 The Interview Mode screen appears. At the Interview Mode prompt, select one of the following options:

```
1 - Simple Interview Mode
2 - Advanced Interview Mode
```

Choose Option 1 to use the default values for certain configuration parameters.

Choose Option 2 to specify all configuration parameters during the interview.

Press `c` to continue.

2 The Interview begins. The prompts will ask you to provide values for the parameters listed in SA Installer Interview Prompts on page 78.

3 If you specified a response file when invoking the Installer, you should accept the parameter defaults as these values are taken from the response file.

If you did not specify a response file, provide values for the parameters as requested in the interview. Certain parameters have SA default values that you can change or accept. Follow the on screen instructions to complete the interview.

The SA Installer displays default values in square brackets [ ]. To accept the default value simply continue to the next interview question.

4 When you have completed the interview, the Installer displays this message:

```
All parameters have values. Do you wish to finish the interview (y/n):
```

If you are satisfied with the parameter values you provided, press y.

If you want to review or change any values, press n. The Installer sequentially displays the parameters again, showing in brackets [ ] the values that you provided and provides a chance to change the values.

After modifying your responses, press y to finish the interview.

5 The installer prompts you to provide a filename for the response file:

```
Name of response file to write
[/usr/tmp/oiresponse.slices_master_typical]
```

All of the parameter values you specified during the Interview will be written to a text response file and saved to the current server at the location you specify. You can enter the full path and name of the response file or accept the default location and name:

`/usr/tmp/oiresponse.slices_master_typical`

Had you specified a Custom Interview, the default file name would be:

`/usr/tmp/oiresponse.slices_master_custom`

Record the fully qualified path to and name of the response file and store it where you can easily find it. You will need to use it again during future installations and upgrades.

6 After saving the response file, you can continue the installation using the response file you just created or end the installation and use the response file later.

```
Would you like to continue the installation using this response file?
(y/n):
```

If you are satisfied with the responses you entered in the interview and you are ready to install SA now, enter `y` to continue.

If you do not want to install SA after completing the interview, enter `n`.

To use this response file later, invoke the Installer with the `-r` option and supply the fully qualified path to the file then go to Phase 3:

```
/opsware_system/opsware_installer/install_opsware.sh -r
<full_path_to_response_file>
```

Phase 3: Install the Core Components

1 The Component Selection screen displays the components that can be installed:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Model Repository, First Core
2 ( ) Core Infrastructure Components
3 ( ) Slice
4 ( ) OS Provisioning Components

Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

The new SA Component Bundle architecture bundles components for installation. Components within a bundle must always be installed together on the same server. For information about how components are distributed in bundles, see **Table 38 on page 105**.

Single Host Installation: If you are installing the Model Repository, Infrastructure Component bundle, Slice Component bundle(s), and optionally, the OS provisioning bundle on a *single server*, press `a` which selects all components and press `c` to continue. The SA Installer begins the installation of the selected SA Core Components.

Distributed Component Installation: If you plan to distribute components on multiple servers, you must run the SA Installer (specifying the response file created in Phase 2, step 5 on page 109) on each server on which you will distribute the components. You must also install the components in the order that they are listed on the *Component Selection* screen (you must install the Model Repository before the Infrastructure Component bundle and the Infrastructure Component bundle before the Slice Component bundle, and so on). In addition, if you plan to install *multiple Slice Component bundles* on the same or different hosts, you must install all of them before installing the OS Provisioning components.

To install the Core Component bundles of an SA Core to different servers, perform the following tasks:

a. Copy the response file created in Phase 2, step 5 on page 109 to all other servers on which you will install Core Component bundles.

b. On each core, copy the contents of the *SA Product Software DVD* to a local disk or network share.

c. On each core, invoke the installer specifying the response file you copied in **step a** above:

```
/opsware_system/opsware_installer/install_opsware.sh -r
<full_path_to_response_file>
```

You must specify the full path name to the response file.

d. The *Component Selection* menu is displayed:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Model Repository, First Core
2 ( ) Core Infrastructure Components
3 ( ) Slice
4 ( ) OS Provisioning Components

Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

e. Select the component(s) to install on that host and press `c` to continue.

During the *Model Repository installation*, the Installer asks if you want to generate a *new database of cryptographic material*, enter `y`.

Phase 4: Post-Component Installation Tasks

1. If you install the Model Repository on a server *without* an installed Slice Component bundle, you must also install a *Server Agent* on that server after Core installation is complete. For more information about deploying SA Agents, see the *SA User Guide: Server Automation*.

2. *If you are distributing Core components*, after the cryptographic material has been generated, copy the `crypto` database and the gzipped Unix tar file from the following directory to every Core Server:

   ```
   /var/opt/opsware/crypto/cadb/realm/opsware-crypto.db.e
   /var/opt/opsware/crypto/cadb/realm/opsware-crypto.tgz.e
   ```

   You must copy the database of cryptographic material and the gzipped tar file to the same directory and must use the same file names on every distributed Core Server. The directory and database must also be readable by the root user.

3. *If you are distributing Core components*, after the Model Repository installation is complete, copy the Oracle `tnsnames.ora` file from the First Core server to all other servers that will host components. Ensure that the path for the file (`/var/opt/oracle/tnsnames.ora`) is the same on all servers. For more information, see tnsnames.ora File Requirements on page 206.

4. *If you are distributing Core Components*, you can install additional instances of the Slice Component bundle which includes a secondary Data Access Engine, Command Center, Core and Agent Gateways, Global file system, Web Services Data Engine, and a Build Manager.

Phase 5: Upload the Software Repository Content

In previous releases, you would have been required in this phase to upload the *OS Provisioning Stage 2 Images* due to certain modifications to Linux installation media that were necessary for compatibility with SA. As of SA 7.80, these modifications are no longer required so there is no longer a requirement to upload OS Provisioning Stage 2 Images. You can continue to use your SA 7.50 Satellites in an SA 7.80 environment, however, the SA 7.50 Satellites will continue to be restricted by the Stage 2 images restrictions limiting which operating systems can be provisioned. After you upgrade your Satellites to SA 7.80, these restrictions will be lifted.

1 Now you must upload the default SA Software Repository Content. On the server where you installed the Software Repository (part of the *Slice Component bundle*), mount the *Agent and Utilities DVD* or NFS-mount the directory that contains a copy of the DVD contents.

The Installer must have *read/write root access* to the directories where it will install the SA components, including on NFS-mounted network appliances.

2 In a terminal window, log in as root and change to the root directory:

```
cd /
```

3 Invoke the Installer with the `-r` (response file) argument. For example:

```
/opsware_system/opsware_installer/install_opsware.sh -r
/usr/tmp/oiresponse.slices_master_typical
```

Specify the fully qualified path to the response file. The directory path in the preceding command assumes that you copied the *SA Agent and Utilities DVD* to a local disk or network share.

The Installer displays following options:

```
Welcome to the Opsware Installer.
Please select the components to upgrade.
1 ( ) Software Repository - Content (install once per mesh)
Enter a component number to toggle ('a' for all, 'n' for none.
```

4 At the install prompt, select `Software Repository`:

```
1 ( ) Software Repository - Content
```

Press `c` to continue. The Installer uploads the Software Repository content.

When the installation of the Software Repository content finishes, the SA installation phase is complete.

You must now complete the tasks in the next section, "Logging in to the SAS Web Client."

Logging in to the SAS Web Client

Now that you have installed an SA Core, completing the following tasks will enable you to log in to the SAS Web Client and begin to create users and user groups and use SA to manage servers in your Facility.

Browser Configuration

To access the Command Center your browser must:

- Accept cookies.
- Have Java™ support enabled.
- Support SSL and provide 128-bit encryption (recommended).
- Have third-party pop-up blockers disabled. As an alternative, use the supported browser's native pop-up blocking. Using a third-party pop-up blocker could prevent certain SAS Web Client functions from working correctly.

Logging in to the SAS Web Client

To log in to the SAS Web Client, perform the following tasks:

1. In a web browser, enter the following URL:

   http://<occ_hostname>

   where *<occ_hostname>* is the host name or IP address of the server on which you installed the Command Center component (part of the Slice Component bundle).

2. Follow the browser's instructions for installing the security certificate.

3. The SAS Web Client will then prompt you for a user name and password.

   Enter `admin` for the user name.

   The password is the SA Admin password you specified during the Installer Interview (`cast.admin_pwd`).

4. When you are logged on, the first task is to create a new *Administrator user.*

   From the Navigation panel, select **Administration ‰ Users & Groups**. Follow the instructions for creating new users in the *SA Administration Guide*

   For the Group Membership, select *SA System Administrators*.

5. Next, create an *Advanced User* using the same method described in **step 4**. For the Group Membership, specify *Advanced Users*.

6. Now log out as `admin` and log back in to the SAS Web Client as the *Advanced User* you created in the previous step. This Advanced User should now be able to use all available SA system functions.

7. Still logged in as the Advanced User, run **System Diagnosis**. From the Navigation panel, click **Administration ‰ System Diagnosis**.

   See the *SA Administration Guide* for detailed information about running and interpreting the output of the System Diagnosis Tool.

Post-Installation Tasks

You must now complete the tasks described in Chapter 7, “First Core Post-Installation Tasks.”

7 First Core Post-Installation Tasks

This section describes system administration tasks that you must perform after installing a First Core.

The SA Client

The SA Client is a powerful Java™ client for the Server Automation System. It provides the look-and-feel of a Microsoft Windows desktop application with the cross-platform flexibility of Java. If you installed your core on multiple servers, you can access the SA Client from any Core Server hosting a Component Slice bundle.

To access the SA Client for the first time, you must invoke the SA Client Launcher from the SAS Web Client Main Page. Clicking on this link will install the SA Client and the required Java Runtime Environment (JRE) on your local machine. Once installed, you can invoke the SA Client from the local machine rather than from the SAS Web Client.

The SA Client is installed with the Java™ 2 Runtime Environment, Standard Edition 1.4.2._15. The SA Client is a Java application that installs and runs with its own Java Runtime Environment (JRE). The SA Client will not interfere with any other versions of JRE you may have installed on your system. The JDK will not be used (and is not usable) by any other Java application on the target computer, and it will not set itself as the default JDK on the target computer.

Note that the SA Client adds certain functionality not in the SAS Web Client. Instructions in this documentation set will explicitly identify either the SAS Web Client or the SA Client as required to complete a task.

See the *SA User Guide: Server Automation* for more information about both clients.

Unattended Installation of the SA Client

This section describes how to perform unattended installation of the SA Client from the command line.

To begin an unattended installation, invoke the installer using the `-q` (quiet) argument, which causes the installer to perform the installation as if you had accepted all default settings and it asks for no user input.

For example, execute the following command on the server on which you want to install the SA Client:

```
opswclientinstaller_windows_1_0.exe -q
```

By default, the SA Client is installed in the following directory:

```
C:\Opsware
```

If you want to install the launcher in another directory, specify the `-d` option, as in the following example:

```
opswclientinstaller_windows_1_0.exe -q -dir C:\Opsware_SAS_Client
```

See the *SA User Guide: Server Automation* for more information on how to use the SA Client.

Adding or Changing an SA Client Launcher Proxy Server

By default, the SA Client uses the proxy server settings configured for the default browser on your local system. For example, if your default browser has no proxy server settings configured, neither will the SA Client.

You can configure SA Client to use a proxy server by editing the Java Web Start `deployment.properties` file.

For details on how to do that, see the *SA User Guide: Server Automation*.

Configuring Contact Information in SA Help

To configure the SA administrator contact information that appears on the SA Help page, perform the following tasks:

1. In the SA Core, log on as `root` to the server running the Command Center (Slice Component bundle).
2. Change to the following directory:

   `/etc/opt/opsware/occ`
3. Open the `psrvr.properties` file in a text editor.
4. Modify the values in the following fields to change the contact information in the SAS Web Client Help:

   ```
   pref.occ.support.href
   pref.occ.support.text
   ```
5. Save the file and exit.
6. Restart the Command Center component by entering the following command:

```
/etc/init.d/opsware-sas restart occ.server
```

Installing Application Configuration (AppConfig) Content

In order to get the baseline set of Application Configurations (AppConfigs) into your core, you must perform the post-installation tasks described in this section using the DCML Exchange Tool (DET).

The AppConfig content archive is located on the Agent & Utilities DVD in the `disk001/packages/` directory:

`OPSWContent-AppConfig-<current_version>.tgz.`

Complete the following steps:

1. The AppConfig content archive is in `tar/gz` format, so you must uncompress it with `gunzip` and extract it using tar. You can also use GNU tar with the `xvzf` flags to simultaneously uncompress and extract the file, for example:

   ```
   tar xvzf OPSWContent-AppConfig-<current_version>.tgz
   ```

This command creates a directory named `AppConfig`.

2. Install the Content Baseline Tool (`cbt`) (for example, `cbt-34_1_0_27.zip`) from the primary Product Software DVD. The tool is located in the directory `/disk001/packages/<core OS>`. Install the tool under `/usr/local` or any known path and add the location to your path, for example:

   ```
   export PATH=$PATH:/usr/local/cbt/bin
   ```

3. Set the JAVA_HOME environment variable to use Opsware's JRE:

   ```
   export JAVA_HOME=/opt/opsware/j2sdk1.4.2/
   ```

4. Verify that `cbt` is working properly by invoking it using `cbt -v`. This command should return a version, if not, check your installation, PATH and/or JAVA_HOME settings.

5. Import the content using a `cbt` config file or by manually entering the user names and passwords for the DCML Exchange Tool and Web Services Data Access Engine users (for example, `admin` and `detuser`):

   ```
   cbt -i AppConfig -cf core.cfg
   ```

   Shown below is a sample `cbt` config file. Change the `*.host` entries and/or passwords as necessary:

   ```
   cbt.numthreads: 5
   mail.from: joeuser@opsware.com
   spike.host: USE YOUR IP ADDRESS FOR YOUR SAS CORE OR COMPONENT
   way.host: USE YOUR IP ADDRESS FOR YOUR SAS CORE
   word.host: USE YOUR IP ADDRESS FOR YOUR SAS CORE
   spin.host: USE YOUR IP ADDRESS FOR YOUR SAS CORE
   twist.host: USE YOUR IP ADDRESS FOR YOUR SAS CORE
   spike.username: admin
   spike.password=admin
   twist.username: detuser
   twist.password=detuser
   ssl.keyPairs: /var/opt/opsware/crypto/twist/spog.pkcs8
   ssl.trustCerts: /var/opt/opsware/crypto/twist/opsware-ca.crt
   twist.certPaths: /var/opt/opsware/crypto/twist/opsware-ca.crt
   ```

6. Launch the SA Client and select **Tools** ‰ **Options** and **Reload cache now**, or wait a few minutes, then verify that your new Content is available.

7. AppConfig content appears in two locations in the SA Client, in the Application Configuration and in the Audit and Remediation feature. To view the AppConfig content in the SA Client, select:

   Navigation pane ‰ **Library** ‰ **By Type** ‰ **Application Configuration**

   or, when viewing an Audit or Snapshot Specification rule:

   Navigation pane ‰ **Library** ‰ **By Type** ‰ **Audit and Remediation**

If you have any questions on any Content, please contact technical support.

SA Agent Discovery and Deployment (ODAD)

The Discovery and Deployment (ODAD) utility allows you to use the SA Client to identify servers on your network that do not have Agents installed and install (deploy) Server Agents onto those servers.

Enabling ODAD for Unix Servers

The HP SA Installer automatically installs all required software to use ODAD with Unix servers during a core installation.

However, before you use ODAD to open remote terminal sessions on unmanaged Unix servers, verify on the server hosting the Agent Gateway (part of the Component Slice bundle), that the `telnet`, `rlogin`, and `ssh` clients reside in either the `/bin`, `/usr/bin`, or `/usr/local/bin` directories. If the client resides in any other directory, create a symbolic link in `/usr/local/bin` to the actual location of the client on your server.

Enabling ODAD for Windows Servers

Before you can use ODAD to deploy Server Agents to Windows servers in a core installation, you must install an agent on a Windows Server that is managed by that core and that is running a 32-bit version of:

- Windows 2000
- Windows 2003
- Windows XP

After the agent is installed, install the *Windows Agent Deployment Helper* on that server.

There must bidirectional connectivity between the server on which the Windows Agent Deployment Helper is installed and the core where the tasks described below will take place.

You can install only one Windows Agent Deployment Helper in each Multimaster Mesh. Note also that, a Windows Agent Deployment Helper will not function properly in an Satellite installation.

To install the Windows Agent Deployment Helper, perform the following tasks:

1. Identify a Windows server on which you can install the Windows Agent Deployment Helper. This server must be running a 32-bit version of Windows 2000, Windows 2003, or Windows XP. (Windows 64-bit operating systems are not supported.)

   On this Windows server, install an agent using the SA Command Line Interface (CLI). For instructions, see the appendix, “SA Agent Utilities”, in the *SA User Guide: Server Automation*.
2. After the Server Agent is installed, log in to the SA Client.
3. From the Navigation pane, select **Devices** ‰ **All Managed Servers**.
4. From the Content pane, select the Windows server on which you installed the agent in **step 1**.
5. From the **Action** menu, select **Attach** ‰ **Attach Software Policy**. The Attach Software Policy window appears.
6. From the list of software policies, select Windows Agent Deployment Helper. (By default, the Remediate Servers Immediately option is selected. Do not deselect this option.)
7. Click **Attach**. The Remediate window appears.
8. Complete the tasks to remediate the server with the Windows Agent Deployment Helper policy. See the *SA User’s Guide: Application Automation* for more information about how to remediate a server using a software policy.
9. Because the SA Client caches information about the Windows Agent Deployment Helper, restart all running SA Clients for this core.

10 Log in as `root` to *all* servers hosting a Component Slice bundle (which contains a Core Gateway) for the core. With a text editor, open the following file:

```
/etc/opt/opsware/opswgw-cgws0-<facility>/opswgw.properties
/etc/opt/opsware/opswgw-cgws1-<facility>/opswgw.properties
```

and so on.

where `cgws0` identifies the first installed Component Slice bundle. Subsequent installed Component Slice bundles will be identified as `cgws1`, `cgws2`, and so on; `facility` is the facility name you applied to the core during installation.

11 Locate the following line:

```
opswgw.IngressMap=${NETBIOSHELPERIP}:NETBIOS
```

12 Replace `${NETBIOSHELPERIP}` with the IP address of the server where you installed the Windows Agent Deployment Helper. For example:

```
opswgw.IngressMap=192.168.165.242:NETBIOS
```

13 If `${NETBIOSHELPERIP}` has already been replaced by an IP address, then the Installer successfully discovered your Windows Agent Deployment Helper and inserted the IP address. You should, however, verify that the automatically discovered IP address is correct.

14 Restart the Core Gateway on each server on which you edited `opswgw.properties` with the following command:

```
/etc/init.d/opsware-sas restart opswgw-cgws
```

Setting up the Windows Agent Deployment Helper When the Administrator Account is Disabled

When, usually for security reasons, the Windows Administrator account is disabled on a Windows server, you must perform the following additional setup tasks to create an account that can run the Windows Agent Deployment Helper:

1 Log in as root to any Unix/Linux server in the same core as the Windows server.

2 Change to the following directory:

```
cd /opt/opsware/oi_util/bin/
```

3 Enter the following command to run the `shared_script_util.sh` script:

```
./shared_script_util.sh modify adt_deploy_agents.bat \
-U NEW_USER -p agentDeployment.deployAgent -e \
-c "Change user name"
```

where the NEW_USER account is a member of the Windows Agent Deployment Helper's local Administrators group. The Windows Agent Deployment Helper can now run under the user name you specified.

4 Enter the following command to review the current script settings:

```
./shared_script_util.sh showpolicy adt_deploy_agents.bat
```

You will see the following output, except that the USER line should contain the name of the account you specified in **step 3**.

```
PTY 0
USER NEW_USER
EXEMPT
PERM agentDeployment.deployAgent
```

Agent Deployment Tool (ADT) Requirements

If you plan to use the Agent Deployment Tool (ADT) to deploy Server Agents, you must have the following in the root user's path on each server hosting the Slice Component bundle(s) (includes the Gateway) and each Satellite server:

- OpenSSH client
- telnet client (standard client that ships with Linux or Solaris)
- rlogin (standard rlogin that ships with Linux or Solaris)

Storage Visibility and Automation

If you plan to use Storage Visibility and Automation, see the *Storage Visibility and Automation Installation & Administration Guide*.

Storage Essentials (SE) version 6.1.1 or later is required to view, report, or perform any Service Automation Visualizer (SAV) and Service Automation Reporter (SAR) operation on SAN objects, such as arrays, switches, volumes, and so on. SAN objects are discovered in Storage Essentials. To enable discovered SAN objects in the SA, SAV, and SAR products, the Server Automation SE Connector component must be installed and configured.

Server Automation Reporting (SAR)

If you plan to use Server Automation Reporting (SAR) you must install or upgrade to SAR 7.80, see the *Server Automation Reporting Installation Guide*, the *SAR 7.80 Upgrade Guide*, and/or the *Live Network Connector 2.2.7 Guide*.

NA/SA Integration

HP Network Automation (NA) Integration with the HP Server Automation (SA) enables IT staff members to see how servers are connected to network devices and to closely examine managed servers. With this information, they can determine how all devices are related and coordinate and implement required changes.

For more information about NA/SA Integration Integration, see the *SA User Guide: Server Automation*.

To set up NA/SA Integration, you must change certain configuration settings in both NA and SA, run diagnostics for NA topology data, and configure user permissions.

To set up NA Integration with the current SA version, you must have Network Automation (NA) 6.1 or later installed.

SA Gateway Requirements

An NA Core can use an existing SA Gateway that was installed for an SA Core, but an SA Core cannot use an existing Gateway that was installed for an NA Core.

Therefore, NA must be configured to use the SA Core's Gateway that was installed using the SA Installer.

SA Client Communication with NA

Ensure that the SA Client can communicate with NA. If the SA Client can't communicate with the NA server, see "Resetting the NA Host" section of the *SA User Guide: Server Automation*.

Edit the jboss_wrapper.conf File

Comment out (or delete) the three lines in `server/ext/wrapper/conf/jboss_wrapper.conf` below:

```
#Following are added for bug 150387

#wrapper.java.additional.6=-Dorg.omg.CORBA.ORBClass=com.sun.corba.se.internal
.Interceptors.PIORB

#wrapper.java.additional.7=-Dorg.omg.CORBA.ORBSingletonClass=com.sun.corba.se
.internal.corba.ORBSingleton
```

#wrapper.java.additional.8=-Xbootclasspath/p:/opt/NA/server/ext/wrapper/lib/CORBA_1.4.2_13.jar

Since SA 7.80 does not use Java 1.4.2, these lines are no longer required.

NA Integration Port Requirements

Before you configure NA Integration, ensure that SA and NA can communicate with each other over the following ports:

- **Port 1032 (NA to SA)**

  NA must be able to access port 1032 on the server that is running the SA Web Services Data Access Engine component (part of the Component Slice bundle). By default, the Web Services Data Access Engine listens on port 1032.

- **Port 8022** (Unix) / Port 22 (Windows) **(SA to NA)**

  For the Global File System (OGFS) feature to be able to display data about network devices, SA must have access to port 8022 (Unix-based NA Servers) and 22 (Windows-based NA Servers).

- **RMI/JRMP Ports for NA API**

  The NA API uses Java RMI to connect to the NA server. SA uses the NA API for the NA integration. RMI/JRMP requires that the following ports are open:

  — **Port 1099**

  JNDI

  — **Port 4444**

  RMI Object

— **Port 8083**

RMI

— **Dynamic**

RMI

See the *NA User Guide* for information about how to set up these port requirements to access the NA API through a firewall.

Time Requirements for NA Integration

The SA and NA core servers must be synchronized and have the same time and time zone settings. See also Time and Locale Requirements on page 65.

Configuring NA for Integration

To set up NA/SA Integration, you must configure NA to use SA Authentication To complete this configuration, you will need to know:

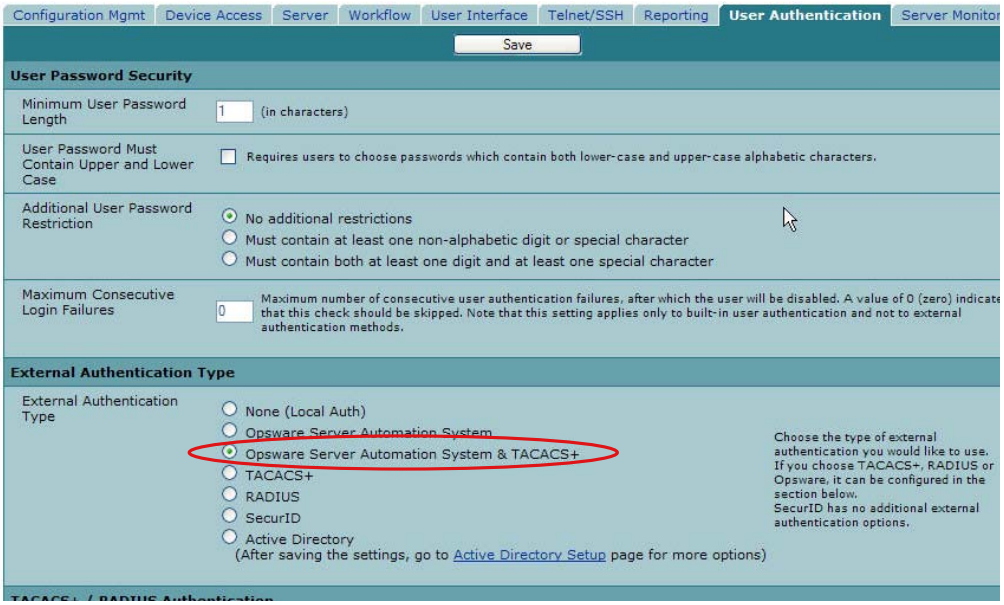
- The IP address or Hostname of the server hosting the Web Services Data Access Engine (part of the Component Slice).
- The port number that the Web Services Data Access Engine listens on.
- The Web Services Data Access Engine user name.
- The Web Services Data Access Engine password
- The IP address or hostname of the server hosting the Command Center.
- The default user group for new SA users.

NA Authentication Configuration

For detailed information on NA/SA authentication, see the *NA User Guide*. To change the authentication settings in NA, perform the following tasks:

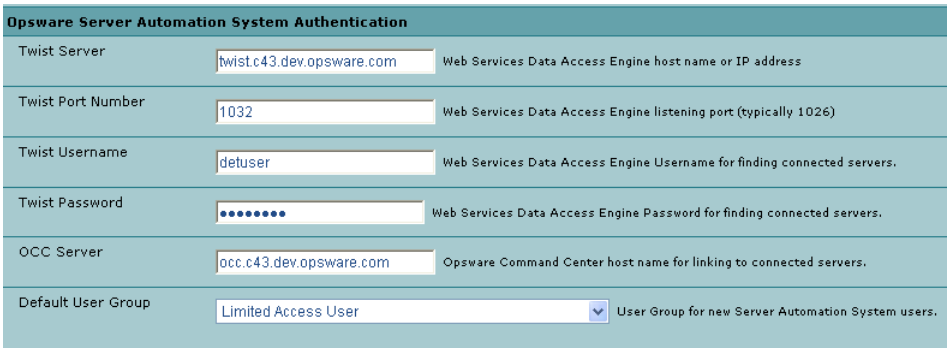
1. Log in to NA.
2. Select **Admin** ‰ **Administrative Settings** ‰ **User Authentication** to display the Administrative Settings — User Authentication page.
3. In the External Authentication Type section, use the radio button to select Opsware Server Automation System & TACACS+ as shown in Figure 13.

Figure 13 External Authentication Type in NA



4 Scroll down and complete all fields in the Opsware Server Automation System Authentication section shown in Figure 14. NA uses the Web Services Data Access Engine Username and Password (specified during installation for the parameter `truth.twistPwd`) when it gathers layer 2 data. NA looks for the server interface information by MAC address, using that user's permissions. The user must have read access to server information.

Figure 14 Opsware Server Automation System Authentication



5 Click **Save** to save your configuration change.

See the *NA User Guide* for more information on NA configuration.

SA Configuration Changes

Complete the following tasks to prepare SA for NA Integration:

- **Specify the NA Server Name and NA Port (Windows) in SA**

  a If you did not specify the NA server name during the SA Installer Interview, you must specify the value for the `twist.nasdata.host=<hostname>` parameter in the `/etc/opt/opsware/twist/twist.conf` file. For more information about modifying this file, see the *SA Administration Guide*.

If you have a multiple slice core, you must edit the `twist.conf` file on all slices. Then, you must restart NA and the Web Service Data Access Engine for each slice.

b If NA is running on a Windows server, you must change the port setting parameter from `nas.port=8022` to `nas.port=22` in the `/etc/opt/opsware/hub/hub.conf` file. A default Windows server installation runs the proxy SSH/Telnet servers on port 22/23 rather than the Unix default of port 8022/8023. See the *NA User Guide* for more information on NA servers.

After you make this configuration change, you must restart the server hosting the Component Slice bundle (specifically for OGFS).

- **Enable the spin.cronbot.check_duplex.enabled Parameter**

  The `spin.cronbot.check_duplex.enabled` parameter must be enabled for NA integration. To do so:

  a Log into the OCC as an SA Administrator.

  b Click on Administration ‰ System Configuration.

  c Select Data Access Engine from the SA component list.

  d Locate the parameter, `spin.cronbot.check_duplex.enabled`.

  e Click `Use value:` and enter `1` in the text box.

  f Click Save.

For more information about using the OCC to modify parameter values, see the *SA Administration Guide*.

Configuring NA/SA Integration with CiscoWorks NCM

If you are deploying SA with CiscoWorks NCM 1.2, you must make certain configuration changes. Some CiscoWorks NCM deployments (where CiscoWorks LMS is co-resident with NCM) use non-standard ports that affect integration with SA.

To determine which changes you will need to make, perform the following tasks:

Phase 1: Edit tomcat4-service.xml:

1 Log in to your NA server.

2 Open the XML file:

`<NAS_install_dir>/server/ext/jboss/server/default/deploy/tomcat4-service.xml`.

3 Search for the string '`scheme=https`"'.

4 Check the preceding entry which should be

`port = "port_no".`

If the `port_no` value is 443, then go to Phase 4; otherwise, note the specified port and continue to Phase 2.

Phase 2: Assign the port number:

1 Log in to the SA Client.

2 In the SA Client, from the **Tools** menu, select **Options**.

3 In the Set Options window, select **Opsware NAS**.

4 In the Host field, append `:<port>` to the hostname, where `<port>` is the port number found in Phase1, **step 4**, for example:

`mycore.opsware.com:443`

Click Save.

The following warning will appear: “General.Host: must be a valid host string.” Ignore this warning. Close the Set Options window.

(Phase 2 must be performed for every user of the SA Client.)

Phase 3: Edit Primary Data Access Engine files:

1 Log in to the SA Core Server where the Primary Data Access Engine is installed (part of the Infrastructure Component bundle).

2 Open the `/opt/opsware/twist/twist.sh` file and change this line:

`https://$NASHOST/tcdocs/truecontrol-client.jar`

to read (assuming that 443 was the port you noted in Phase 1, **step 4**):

`https://${NASHOST}:443/tcdocs/truecontrol-client.jar`

3 Restart the server hosting the Web Services Data Access Engine (part of the Component Slice bundle):

`/etc/init.d/opsware-sas restart twist`

(You will need to perform Phase 3 for each Web Services Data Access Engine server installation.)

Phase 4: Assign the SSH port:

1 Log in to NA.

2 Select **Admin** ‰ **Administrative Settings** ‰ **Telnet/SSH** to display the Administrative Settings - Telnet/SSH page.

3 In the SSH Server section, locate the SSH Server Port.

4 If the port is 8022, then you are finished; otherwise, note the port being used and continue to Phase 4, **step 5**.

5 Log in to the SA Core Server where the Global File System (OGFS) is installed (part of the Slice Component bundle).

6 Open the `/etc/opt/opsware/hub/hub.conf` file and change the value for `nas.port` to the port you found in Phase 4, **step 4**. For example:

`nas.port=9022`

Topology Data

You must also run the NA Topology Data Gathering and NA Duplex Data Gathering diagnostics. For instructions, see the *NA User Guide*.

User Permissions for NA Integration

Access permissions for NA/SA Integration are based on two separate databases: a NA database and a SA database. NA uses its own database for authorization. SA uses a different security mechanism for authorization. However, for NA integration, all authentication (for both NA and SA) is processed by SA.

When NA is configured to use SA authentication, it tries to authenticate against SA first. If NA fails to authenticate against SA, it falls back to the NA database. If there is an account in the NA database, the fallback is only allowed if that user is configured to allow fallback authentication. See the *NA User Guide* for more information on NA authentication.

When a new user is authenticated through SA, an account is created in NA. The account is placed in the Default User Group that was specified when SA authentication was enabled in the Administrative Settings in NA. This user group, which is configurable, controls the default permissions that the system administrator has assigned to SA users.

You must have the required set of permissions to view servers and network devices. To obtain these permissions, contact your SA administrator, or for more information, see the *SA Administration Guide*.

Operations Orchestrator/SA Integration

To configure the Operations Orchestrator (OO)/SA connector in an SA 7.5 multi-slices core with multiple Command Engine servers, you must do the following in *each Command Engine host in the core*:

1 Create/update the connector configuration file located in:

```
/etc/opt/opsware/iconclude-connector/iconclude.conf
```

Supply the values appropriate to your OO/SA installation:

```
# Copy this to /etc/opt/opsware/iconclude-connector/
# iconclude.conf and provide appropriate values
iconclude.enabled:1 or 0
#1 = enabled, 0 = disabled
iconclude.host:<hostname or IP>
# hostname/IP of OpsForce Central server
iconclude.port:<port_number>
# port of OpsForce Central server, for example, 8443
iconclude.proto: <protocol>
# Protocol to use, valid values, http or https
iconclude.flow.approve:Library/My Ops Flows/SAS_Integration/
SAS_Job_Approval_Integration
# flow to use for requesting approval
iconclude.user: <username>
# iConclude username
iconclude.password:<password>
```

2 (**Optional**) To encrypt the Operations Orchestrator user password, do the following:

a `mkdir -p /var/opt/opsware/crypto/iconclude-connector/`

b `echo -n "secret" > /var/opt/opsware/crypto /iconclude-connector/iconclude.pwd`

c `chmod -R go-rwx /var/opt/opsware/crypto /iconclude-connector`

DHCP Configuration for OS Provisioning

The Dynamic Host Configuration Protocol (DHCP) specifies how to assign dynamic IP addresses to servers on a network. OS Provisioning uses DHCP to allow network booting and configuration of unprovisioned servers in the Server Pool. DHCP is also used to configure networking on newly provisioned servers that have not been assigned a static network configuration.

For OS provisioning, you may use either the DHCP server included with SA, an existing ISC DHCP server, or the MS Windows DHCP server. The instructions for configuring these various DHCP servers are in the following sections:

- Configuring the SA DHCP Server for OS Provisioning
- Configuring an Existing ISC DHCP Server for OS Provisioning
- Configuring the Windows DHCP Server for OS Provisioning
- Controlling the SA and Windows DHCP Servers Responses to OS Provisioning Requests

DHCP Software included with the Boot Server

When you install the Boot Server, the SA Installer also installs the following:

- **dhcpd**: An Internet Software Consortium DHCP server (ISC dhcpd).
- **dhcpd.conf**: A default DHCP server configuration file, read by the `dhcpd` server.
- **dhcpdtool**: The SA DHCP Network Configuration Tool which allows you to modify the `dhcpd.conf` file.

SA DHCP Server (dhcpd)

The DHCP server provides service to two types of networks:

- **Local networks**: Networks that are attached directly to the network interfaces of the host running the DHCP server. No special network configuration is needed to support local networks.
- **Remote networks**: Networks that are not directly attached to the DHCP server host. A router sits between the DHCP server host and the remote networks. For remote networks, a DHCP proxy (sometimes called IP helper) must be configured on each remote network to relay DHCP packets to the DHCP server host.

A DHCP proxy is not provided with SA and instructions for setting one up are beyond the scope of this document. DHCP proxy functionality is often included in modern routers. Check with your network administrator or router vendor.

Log messages that the DHCP server produces are sent to the standard Unix syslog process with the daemon facility. Consult your vendor documentation on how to configure and view syslog messages.

See "Starting and Stopping the SA DHCP Server" on page 130.

SA dhcpd.conf File

The `dhcpd.conf` file provides the necessary parameters to support network booting of Sun hardware (a DHCP-capable PROM is required) and x86 hardware (a PXE-compatible system is required).

For x86 hardware that does not support PXE, the server can be booted from a floppy (Windows) or CD (Linux). When a boot floppy or CD is used, the DHCP server still provides network configuration information to the host.

The DHCP configuration file is `/etc/opt/opsware/dhcpd/dhcpd.conf`. In most cases, you will modify this file by running the DHCP Network Configuration Tool. For some advanced configurations (as noted in the following section), you may need to modify the file with a text editor. Documentation on the DHCP configuration file is available at the ISC web site `www.isc.org`.

The DHCP leases file is `/var/opt/opsware/dhcpd/dhcpd.leases`. This file should not need editing.

SA DHCP Network Configuration Tool (dhcpdtool)

The DHCP Network Configuration Tool is a menu-driven, terminal-based utility that enables you to customize the `dhcpd.conf` file for common local and remote network configurations. The tool prompts you for network information needed to configure DHCP for each OS provisioning network. Using the DHCP Network Configuration Tool simplifies configuration of the DHCP server and ensures that the DHCP configuration contains the options that are needed for the OS Provisioning feature to function properly.

If you need to configure the network for OS Provisioning to support less common configurations, you must modify the `dhcpd.conf` file with a text editor. Less common configurations include dual-interfaces with split-horizon DNS requirements, private build networks, and static NAT. Contact Technical Support for more assistance.

Additionally, in some environments, multiple IP networks (layer 3) are layered on top of a single VLAN (layer 2). While this configuration is supported by the ISC DHCP server, generally such a topology requires careful consideration to work properly with DHCP. Therefore, the DHCP Network Configuration Tool can only configure a single IP network per VLAN.

The man pages for the DHCP Network Configuration Tool are installed in `/opt/opsware/dhcpd/man` on the Boot Server. They are also available at the Support web site.

Required Information for the SA DHCP Network Configuration Tool

Before you use the DHCP Network Configuration Tool to configure an OS provisioning network, you need the following information:

- The range of IP addresses that are assigned dynamically by the DHCP server. For example, 192.168.0.11 - 192.168.0.20 might be used to configure a pool of 10 addresses.

Each of these IP addresses must resolve to a host name on the DNS server.

- The IP addresses of one or more DNS servers. The servers must be able to resolve the standard required SA DNS entries. The DNS servers do not need to be on the same network that is being configured.
- A default DNS domain. This domain must include the standard, required SA DNS entries. For example, if the default DNS domain is `example.org`, then there must be an entry `spin.example.org` that can be resolved by the DNS servers.

If you are going to configure a remote network with the DHCP Network Configuration Tool, you will also need to provide the following information:

- The network address and size (netmask or bits). For example, 192.168.0.0/255.255.255.0 or 192.168.0.0/24. Both specify a network range of 192.168.0.0 - 192.168.0.255.
- The network gateway or default router, for example, 192.168.0.1.

Configuring the SA DHCP Server for OS Provisioning

The DHCP Network Configuration Tool is installed with the Boot Server. Perform the following steps to configure networks for OS provisioning:

1. Log in as root to the server running the Boot Server.
2. Make a backup copy of the configuration file with the following commands:

   ```
   cd /etc/opt/opsware/dhcpd
   cp dhcpd.conf dhcpd.conf.orig
   ```

3. Run the DHCP Network Configuration Tool with the following command:

   ```
   /opt/opsware/dhcpd/sbin/dhcpdtool
   ```

   The following DHCP Network Configuration Tool main menu appears:

Example: DHCP Network Configuration Tool Main Menu

```
Opsware DHCP Network Configuration Tool

 a)dd a new network.
 e)xit.

Choice [a, e]:
```

4. To add a new network, enter `a` at the preceding prompt.

   The following menu to add local or remote networks appears:

Example: Menu to Add Local or Remote Networks

```
Opsware DHCP Network Configuration Tool

You may view/edit/delete one of the currently configured network(s):

 1) 192.168.164.0/28
 2) 192.168.165.128/28

Or

 a)dd a new network.
 e)xit.

Choice [1..2, a, e]: a:
```

5. To configure the DHCP service on the local network, enter `l` at the preceding prompt. Local networks are detected automatically and displayed,

   or,

   to add a remote network, enter `r` at the preceding prompt.

6. If you are adding a local network, you need to enter the IP addresses or host names of the DHCP range and the DNS servers.

   In the following example, note that the IP addresses are separated by a comma and a space.

Example: Local Network Configuration

```
Opsware DHCP Network Configuration Tool

Editing DHCP information for 192.168.8.0/23 (255.255.254.0)

All values which prompt for an address accept either a IP or a hostname.

Enter the DHCP Range (start address, stop address)
: 192.168.8.20, 192.168.8.29
Enter the DNS server(s) (comma separated)
: 192.168.2.25, 192.168.2.28
Enter the DNS domain: opsware.com
```

7 If you are adding a remote network, supply information for the network address, size, and gateway. See the following example:

Example: Remote Network Configuration

```
Opsware DHCP Network Configuration Tool

All values which prompt for an address accept either a IP or a hostname.

Enter network/netmask or network/bits: 192.168.10.0/24
Enter the network gateway: 192.168.10.1
Enter the DHCP Range (start address, stop address)
: 192.168.10.51, 192.168.10.59
Enter the DNS server(s) (comma separated)
: 192.168.2.25, 192.168.2.28
Enter the DNS domain: opsware.com
```

8 If the displayed information is correct, enter k to keep the network and return to the main menu.

9 At the main menu, to save the information you have entered, enter s,

or,

to edit a configured network, enter the corresponding integer and go back to step 3,

or,

to add more networks, enter a and go back to step 3.

10 To exit the DHCP Network Configuration Tool, enter e. You are prompted to start (or restart) the DHCP server process.

11 To start (or restart) the DHCP server process, enter y. The DHCP Network Configuration Tool displays diagnostic output as part of its startup.

Starting and Stopping the SA DHCP Server

To start the DHCP server process, enter the following command on the server running the Boot Server:

```
/etc/init.d/opsware-sas start dhcpd
```

To stop the DHCP server process, enter the following command on the server running the Boot Server:

```
/etc/init.d/opsware-sas stop dhcpd
```

Modifying the dhcp.conf File for Use with WINPE

Typically, networks with other DHCP servers do not have the SA DHCP server configured as authoritative. However, WINPE requires that the DHCP server be authoritative in order for servers to be able to boot using WINPE.

The `dhcp.conf` file provided with SA 7.80 by default has the `authoritative` setting commented out. If you need to boot servers using WINPE, you will need to uncomment this line:

1. Log on to the Core's dhcpd server as root.
2. cd to the `/etc/opt/opsware/dhcpd` directory.
3. Issue the following command:

   ```
   chmod a+w dhcpd.conf
   ```

4. Open an editor and edit the dhcp.conf file, for example:

   ```
   vi dhcpd.conf
   ```

   and add or uncomment `authoritative`.
5. Save the file
6. Issue the command:

   ```
   chmod a-w dhcpd.conf
   ```

7. Issue the command:

   ```
   /etc/init.d/opsware-sas restart dhcpd
   ```

Configuring an Existing ISC DHCP Server for OS Provisioning

You may use an existing ISC DHCP server for OS provisioning instead of the DHCP server included with SA. An existing ISC DHCP server will work with the provisioning of PXE 2.0 clients, but not with older clients such as PXE 0.99 or 1.0. (These older PXE clients have old PROMS and a PXE bootstrap floppy made with `rbfg.exe`.) The following instructions apply to recent versions of an ISC DHCP server, such as version 3.02rc3.

To configure an existing ISC DHCP server, perform the following steps:

1. The SA DHCP server must not be running on the server hosting the Boot Server. To disable DHCP on that server:

   On a Linux server, enter the following command:

   ```
   chkconfig --level 345 dhcpd off
   ```

   On a Solaris server, enter the following commands:

   ```
   rm /etc/rc2.d/S90dhcpd
   rm /etc/rc0.d/K30dhcpd
   ```

2. Ensure that the configuration file for the existing ISC DHCP server has the entries shown in: Sample Configuration File Entries for an Existing ISC DHCP Server on page 132.

   The example is a snippet of the `dhcp.conf` file shipped with SA, with the addition of `next-server`. This addition tells the PXE client to look for the `tftpserver` on the SA Core, not on the existing DHCP server.

If you copy and paste the example, change all of the IP addresses (`1.2.3.4`) to the IP address of your core.

3 Ensure that the DHCP scope for the systems to be provisioned is set up with the required details, such as the DNS server, netmask, default router, DNS domain, and so forth.

4 Restart the existing ISC DHCP server.

Sample Configuration File Entries for an Existing ISC DHCP Server

```
#
# declare OPSW site options
#
option space OPSW;
#
# DANGER WILL ROBINSON - if you change the codes for these
# options, you'll need to also edit them in the param-request-
# lists appearing below. Note that in the pxeclient section, you # need to
specify the values in hex, not in decimal. Also, these
# values are burned into a couple other files you'll need to
# edit as well:
# /opt/opsware/boot/tftpboot/pxelinux.cfg/default
# /opt/opsware/boot/jumpstart/Boot/etc/dhcp/inittab
# /opt/opsware/boot/jumpstart/Boot/etc/default/dhcpagent
#
option OPSW.buildmgr_ip code   186 = ip-address;
option OPSW.buildmgr_port code 187 = unsigned integer 16;

#
# define OPSW site options
#
site-option-space "OPSW";
option OPSW.buildmgr_ip   1.2.3.4;
 option OPSW.buildmgr_port 8017;

#
# declare SUNW jumpstart vendor options (Sun recommended naming)
#
option space SUNW;
option SUNW.SrootIP4 code 2  = ip-address;
option SUNW.SrootNM  code 3  = text;
option SUNW.SrootPTH code 4  = text;
option SUNW.SbootFIL code 7  = text;
option SUNW.SinstIP4 code 10 = ip-address;
option SUNW.SinstNM  code 11 = text;
option SUNW.SinstPTH code 12 = text;
option SUNW.SsysidCF code 13 = text;
option SUNW.SjumpsCF code 14 = text;
option SUNW.Sterm    code 15 = text;

#
# define SUNW jumpstart vendor options
#
class "solaris-sun4u" {
   match option vendor-class-identifier;
   vendor-option-space SUNW;
   next-server 1.2.3.4;
   option SUNW.SrootIP4 1.2.3.4;
   option SUNW.SrootNM  "js";
```

```
   option SUNW.SrootPTH "/opt/opsware/boot/jumpstart/Boot";
   option SUNW.SinstIP4 1.2.3.4;
   option SUNW.SinstNM  "js";
   option SUNW.SjumpsCF "js:/opt/opsware/boot/jumpstart/Conf";
   option SUNW.SsysidCF "js:/opt/opsware/boot/jumpstart/Conf";
   option SUNW.Sterm    "vt100";
   option SUNW.SbootFIL "/platform/sun4u/kernel/sparcv9/unix";
#
# We use a bogus install path just to give the installer
# something to mount for now.
#
   option SUNW.SinstPTH "/opt/opsware/boot/jumpstart/Boot";
   option dhcp-parameter-request-list 1,3,6,12,15,43,186,187;
}
#
# Begin dhcptool added SUNW client classes (do not edit)
#
subclass "solaris-sun4u" "FJSV.GPUU";
subclass "solaris-sun4u" "NATE.s-Note_737S";
subclass "solaris-sun4u" "NATE.s-Note_747S";
subclass "solaris-sun4u" "NATE.s-Note_777S";
subclass "solaris-sun4u" "SUNW.Netra-T12";
subclass "solaris-sun4u" "SUNW.Netra-T4";
subclass "solaris-sun4u" "SUNW.Sun-Blade-100";
subclass "solaris-sun4u" "SUNW.Sun-Blade-1000";
subclass "solaris-sun4u" "SUNW.Sun-Fire-15000";
subclass "solaris-sun4u" "SUNW.Sun-Fire-280R";
subclass "solaris-sun4u" "SUNW.Sun-Fire-480R";
subclass "solaris-sun4u" "SUNW.Sun-Fire-880";
subclass "solaris-sun4u" "SUNW.Sun-Fire";
subclass "solaris-sun4u" "SUNW.Ultra-1-Engine";
subclass "solaris-sun4u" "SUNW.Ultra-1";
subclass "solaris-sun4u" "SUNW.Ultra-2";
subclass "solaris-sun4u" "SUNW.Ultra-250";
subclass "solaris-sun4u" "SUNW.Ultra-30";
subclass "solaris-sun4u" "SUNW.Ultra-4";
subclass "solaris-sun4u" "SUNW.Ultra-5_10";
subclass "solaris-sun4u" "SUNW.Ultra-60";
subclass "solaris-sun4u" "SUNW.Ultra-80";
subclass "solaris-sun4u" "SUNW.Ultra-Enterprise-10000";
subclass "solaris-sun4u" "SUNW.Ultra-Enterprise";
subclass "solaris-sun4u" "SUNW.UltraAX-MP";
subclass "solaris-sun4u" "SUNW.UltraAX-e";
subclass "solaris-sun4u" "SUNW.UltraAX-e2";
subclass "solaris-sun4u" "SUNW.UltraAX-i2";
subclass "solaris-sun4u" "SUNW.UltraSPARC-IIe-NetraCT-40";
subclass "solaris-sun4u" "SUNW.UltraSPARC-IIe-NetraCT-60";
subclass "solaris-sun4u" "SUNW.UltraSPARC-IIi-Engine";
subclass "solaris-sun4u" "SUNW.UltraSPARC-IIi-Netract";
subclass "solaris-sun4u" "SUNW.UltraSPARC-IIi-cEngine";
subclass "solaris-sun4u" "SUNW.UltraSPARCengine_CP-20";
subclass "solaris-sun4u" "SUNW.UltraSPARCengine_CP-40";
subclass "solaris-sun4u" "SUNW.UltraSPARCengine_CP-60";
subclass "solaris-sun4u" "SUNW.UltraSPARCengine_CP-80";
#
```

```
# End dhcptool added SUNW client classes (do not edit)
#
# declare PXE vendor options
#
option space PXE;
option PXE.mtftp-ip             code 1  = ip-address;
option PXE.mtftp-cport          code 2  = unsigned integer 16;
option PXE.mtftp-sport          code 3  = unsigned integer 16;
option PXE.mtftp-tmout          code 4  = unsigned integer 8;
option PXE.mtftp-delay          code 5  = unsigned integer 8;
option PXE.discovery-control    code 6  = unsigned integer 8;
option PXE.discovery-mcast-addr code 7  = ip-address;
option PXE.boot-item            code 71 = unsigned integer 16;

#
# define PXE vendor options
#
class "pxeclients" {
   match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
   vendor-option-space PXE;
   filename "pxelinux.0";
   next-server 1.2.3.4;
   option vendor-class-identifier "PXEClient";
#
# We set the MCAST IP address to 0.0.0.0 to tell the boot ROM we
# can't provide multicast TFTP, so it will have to use just
# plain ol' TFTP instead (address 0.0.0.0 is considered
# as "no address").
#
   option PXE.mtftp-ip 0.0.0.0;
   option dhcp-parameter-request-list =
   concat(dhcp-parameter-request-list,ba,bb);
}
```

Configuring the Windows DHCP Server for OS Provisioning

You can use a Microsoft Windows DHCP server instead of the Opsware-supplied DHCP server to provision both Windows or Linux on PXE 2.0 clients.

The Microsoft Windows DHCP server *cannot* be used during the OS provisioning of the following types of systems:

- Solaris
- PXE 0.99 or 1.x clients (These older PXE clients have old PROMS and a PXE bootstrap floppy made with `rbfg.exe`.)

To configure a Microsoft Windows DHCP server for use with OS Provisioning, perform the following tasks:

1. On the Windows system running the DHCP server, you must define option #60, so that it appears in the DHCP scope options. To do so, open a command prompt window, and enter the following command:

   ```
   netsh.exe dhcp server add optiondef 60 "PXEClient" STRING
   ```

2. Using the Windows DHCP Management Snap-in (`dhcpmgmt.msc`), create a scope, which is usually a subnet declaration. In the scope options, #60 should now appear. Check the box, and then add the string `PXEClient`.

3 Using the same scope options box, configure options 66 and 67: Click the DHCP option #66 (Boot Server Host Name), and add the full DNS name of the TFTP/Boot Server (for example `core01.test.com`). For option #67 (Bootfile Name), add the boot file name: `pxelinux.0`.

4 Ensure that the DHCP scope for the systems to be provisioned is configured with the required details, such as the DNS server, netmask, default router, DNS domain, and so on.

5 At the command prompt, enter the following commands to define the IP address of the Agent Gateway and the port forward for the Build Manager:

```
netsh.exe dhcp server add optiondef 186 "buildmgr_ip" IPADDRESS

netsh.exe dhcp server add optiondef 187 "buildmgr_port" WORD
```

6 Using the DHCP Management Snap-in (`dhcpmgmt.msc`), configure options 186 and 187 to be part of your scope, and give them the appropriate values (IP address of the Agent Gateway and the port forward for the Build Manager, normally 8017).

7 Define option 043 (Vendor specific options) as a BINARY type, with the value `01 04 00 00 00 00 ff`. This setting tells the DHCP server to go directly to the FTP server specified in the Boot Server Host Name parameter, and also tells it to not use Multicast TFTP.

8 Restart the Windows DHCP server.

Controlling the SA and Windows DHCP Servers Responses to OS Provisioning Requests

You can configure the SA DHCP server to respond only to the OS provisioning requests from PXE and Sun Solaris JumpStart clients while the Microsoft Windows DHCP server responds to all Windows provisioning requests.

1 Add the network subnet to the SA DHCP server. See "Configuring the SA DHCP Server for OS Provisioning" on page 129.

2 Stop the SA DHCP server:

```
/etc/init.d/opsware-sas stop dhcpd
```

3 Make a backup copy of the SA DHCP configuration file:

```
cd /etc/opt/opsware/dhcpd
cp dhcpd.conf dhcpd.conf.orig
```

4 In a text editor, open the SA DHCP configuration file.

5 Below the `pool` entry, find the subnet definition you want to configure and comment it out with the # character:

```
range <IP1> <IP2>;
```

Should now read:

```
# range <IP1> <IP2>;
```

6 Immediately after the now commented out range line, enter:

```
pool {
      # range <IP1> <IP2>;
      allow members of "solaris-sun4u";
      allow members of "solaris-sun4us";
      allow members of "pxeclients";
      range <IP1> <IP2>;
      }
```

modifying the above as necessary to fit your system. The `pool` statement tells the DHCP server to continue serving the specified range, but only for the three types of clients indicated. (The first two `allow` statements are for Sun machines, the third is for PXE clients). The closing brace in the `pool` statement is required.

7 Repeat the preceding two steps for every subnet you wish to configure.

8 In the text editor, save the `dhcpd.conf` file.

9 Start the SA DHCP server:

```
/etc/init.d/opsware-sas start dhcpd
```

10 Check the DHCP logs for errors. The DHCP service logs with `syslog`. See the `syslog.conf` file to determine how logging has been configured for the SA DHCP server.

11 Ensure that the Windows DHCP server subnet/scope declarations are modified to include the Build Manager DHCP options (code 186 and 187). See "Configuring the Windows DHCP Server for OS Provisioning" on page 134.

12 Ensure that the Windows DHCP server does not include options 43, 60, 66, or 67 in the scope/subnets. This will prevent the PXE and Sun JumpStart clients from connecting to the Windows DHCP server but allow them to connect to the SA DHCP server.

13 Ensure that the IP ranges of the Windows and SA DHCP servers don't overlap. As a guideline, the number of IP addresses in a given range should be twice the maximum number of servers that will be provisioned concurrently.

14 If the DHCP servers aren't directly connected to the network/subnet of the systems being provisioned, the DHCP requests must be forwarded to both DHCP servers, the SA DHCP server first.

Additional Network Requirements for OS Provisioning

OS Provisioning for Solaris

If you are using OS provisioning for Solaris (JumpStart) on an isolated network, you must have a default Gateway (router) available, even if it does not route packets. For Solaris JumpStart to function properly, the IP address of the default Gateway must be sent to the installation client that is being provisioned with DHCP. When you use the SA DHCP Configuration Tool, a default Gateway is properly configured for Solaris because the tool adds the appropriate default router.

Host Name Resolution

For Windows OS provisioning, the host name `buildmgr` must resolve on all Windows OS installation clients.

The SA Core host names must resolve using the DNS search order and DNS server information that the DHCP server provides. The DHCP server provides the DNS server IP address and the DNS search order. For each subnet you configure with the SA DHCP Configuration Tool, the DNS domain used by that subnet must have a DNS entry for `buildmgr`.

For example, you could have two subnets with the following domain names:

```
subnet1.example.com
subnet2.example.com.
```

Therefore, there must be two DNS entries for `buildmgr`:

```
buildmgr.subnet1.example.com
buildmgr.subnet2.example.com.
```

The host running the OS Provisioning Media Server must be able to resolve the IP address to the host name (reverse lookup) for any server being provisioned.

See also Host and Service Name Resolution Requirements on page 60.

Open Ports

Any server on which an OS is to be provisioned must meet the same requirements for connectivity to the SA Core network as any managed server. See "Open Ports" on page 59.

Windows Patch Management Tasks

This section includes post-installation tasks for the SA Windows Patch Management feature.

Import Windows Patches into the Software Repository

Before Windows patches can be installed on managed servers using SA, the patches must be imported into the Software Repository. You can import the patches with the SA Client or with the following shell script:

```
/opt/opsware/mm_wordbot/util/populate-opsware-update-library
```

This script downloads the Microsoft Patch Database and patches from the Microsoft site and imports them into the Software Repository. You should schedule the script to run weekly as a `cron` job on the Software Repository server. Non-administrative users of the SA Client will have the new patches available to them without any action on their part.

For more information about the Opsware-supplied Windows Patch Import script, see the *SA Administration Guide*". For more information about importing Windows patches using the SA Client, see the *SA User's Guide: Application Automation*.

Install Internet Explorer 6.0 or Later for Patch Management on Windows NT 4.0 and Windows 2000

The `mbsacli.exe` patch utility for patch management on Windows NT 4.0 and Windows 2000 requires Internet Explorer 6.0 or later. Note that IE 6.0 is pre-installed on Windows Server 2003.

Automating Installation of IE 6.0 or Later

To automatically deploy IE 6.0 or later, use the Internet Explorer Administrator's Kit (IEAK) for the version of IE that you want to install. For more information on IEAK, see the following URL:

```
http://microsoft.com/windows/ieak/default.asp
```

To automate deployment of IE 6.0 or later to managed servers, perform the following tasks:

1. Install IEAK on a Windows 2000 or Windows Server 2003 system.
2. After you install IEAK, start the Internet Explorer Customization Wizard.

3. IEAK will prompt you to choose a Media Selection option. Select the option *Flat* (all files in one directory).
4. Accept the defaults for all other options.
5. After the wizard completes, zip the contents of the directory it created. This directory contains the automatically deployable version of IE 6.0 or later.
6. Upload the ZIP package into the SA Software Repository. See the *SA Policy Setter Guide* for instructions on importing software into the Software Repository.

   Set the following properties for the package when you import it into the Software Repository. See the *SA Policy Setter Guide* for the steps to edit the properties for a package in the SA Client.

   - In the Installation Parameters section in the **Install Flags field**, specify the installation location:

     `%SystemDrive%\IE-redist`

   - In the Installation Parameters section in the **Reboot Required field**, specify Yes.
   - In the Install Scripts section in the **Post-Install Script tab**, enter this text:

     `%SystemDrive%\IE-redist\ieX.xsetup.exe /q:a /r:n`

   Where `ieX.xsetup.exe` is the IE stub installer and `X.x` identifies the version.

   The `/q:a` install option specifies quiet install mode, with no user prompts. The `/r:n` install option suppresses restarting the server after IE installation.
7. Start the SAS Web Client, create a Software Policy, and add the package you imported into the Software Repository in **step 6** to that policy. See the *SA Policy Setter Guide* for the steps to create a software policy and add a package to a software policy.
8. Use the SA Client to remediate the Software Policy to your managed Windows servers. See the *SA User's Guide: Application Automation* for the steps to install software on a server by remediating a software policy onto a managed server.

Support for Red Hat Network Errata and Channels

The Red Hat Network (RHN) is a web-based system for administrators that assists them in patch management, updating, monitoring, and maintenance. Of particular interest to SA administrators is the ability to install and upgrade packages (RPMs) on Red Hat Linux servers.

Included with SA, the `rhn_import` CLI program allows you to download packages from the Red Hat Network, upload the packages into SA Software Repository, and create software policies that correspond to Red Hat Network patches, errata, and channels. When you remediate the software policies, the packages in the policies are installed or upgraded on the managed servers.

SA administrators can import these packages and create software policies using the SA Client. Alternatively, all these operations can be done from the command line using the `rhn_import` utility. This remediation process can be transparent to end users.

For more information on `rhn_import`, see "Automatically Importing Red Hat Network Errata" in the *SA Policy Setter Guide*.

Global File System Tasks

This section contains optional post-installation tasks for the Global File System (OGFS).

Configuring User ID Numbers for the Global File System

When you install a SA Core, you can set values to control the range of UID and GID numbers used by the Global File System. These values are used to provide unique user IDs for all SA users that are logged in to the OGFS. When the Web Services Data Access Engine creates a new user, it will use these values to determine the next available (unique) user ID that is within the range for the local data center.

To set values that control the range of UID and GID numbers, you must specify the following Web Services Data Access Engine parameters in the `params.conf` file:

- **twist.min_uid**: Contains the minimum UID number that can be used. The default value is 80001.
- **twist.default_gid**: Contains the group ID number that a user is assigned to restrict SA users from using certain ports. The default value is 70001.

These parameters are specified as global in the `params.conf` file, which means that they will be written out to the global response file (`oiresponse.global`). This file is generated when the Model Repository export is performed on the First Core server. When you follow the installation instructions and provide the global response file (`oiresponse.global`) as the initial response file to the Secondary Core server, SA Installer will use the specified values.

For more information, see Table 34, “Global File System Prompts,” on page 94.

After you make changes to these parameters, you must restart the Web Services Data Access Engine server.

8 Multimaster Mesh Installation

This section describes how to run the SA Installer to create a Multimaster Mesh of SA Cores by adding additional cores to the mesh. These instructions are followed by a short list of post-installation tasks.

Multimaster Mesh Installation Basics

A *Multimaster Mesh* is a set of two or more SA Cores that communicate through Management Gateways and can perform real-time synchronization of the data about their Managed Servers contained in their respective Model Repositories over the network. The first SA core installed in any Facility is Multimaster-ready and is called the *First Core*. The second, third, or subsequent cores that you install in that Facility are *Secondary Cores* and along with the First Core can form a *Multimaster Mesh*.

The Model Repositories in each of the cores of this Multimaster Mesh are continually updated so that they are always exact duplicates of each other. All the servers in a Multimaster Mesh can be managed through a single SAS Web Client. A Multimaster Mesh is best for larger networks that span multiple facilities.

The SA Core Component that propagates and synchronizes changes from each model repository database to all other model repository databases is called the *Model Repository Multimaster Component* and provides replication capabilities. This replication capability allows you to store and maintain a blueprint of software and environment characteristics for each facility making it easy to rebuild your infrastructure in the event of a disaster. It also provides the ability to easily provision additional capacity, distribute updates, and share software builds, templates and dependencies across multiple facilities — all from a single user interface.

The following procedures assume that you have already installed the First Core. If not, follow the installation procedures described in Chapter 6, “Installing the First Core” to install the First Core.

Prerequisites for Multimaster Mesh Installations

This section discusses prerequisites for installation and pre-existing conditions that might affect your Multimaster installation.

The First Core

Before adding subsequent Secondary Cores to a Multimaster Mesh, you must have installed the First Core as described in Chapter 6, “Installing the First Core”. You can then perform the tasks in this section to install subsequent cores in the mesh.

First Core Response File oiresponse.slices_master_typical

During the installation of the First Core for this mesh, a response file was created after the interview during the installation of the First Core components. The default response file name is `oiresponse.slices_master_typical`. You will need to copy this file to the server(s) that will host the Secondary Core(s) in order to complete the installation of any Secondary Core in this mesh.

Command Center (OCC)

The Command Center OCC is bundled into the Slice Component bundle. Any Core Server in your Multimaster Mesh that has a Slice Component bundle installed will have the Command Center (OCC) component installed. All servers, First Core or Secondary Core, with the OCC component installed can be used to manage the servers in the Facility that the server is associated with.

Plan Your Core Deployment

You must plan your SA system deployment. You must decide whether you want to install the Core Components on a single server or on multiple servers, whether you will have multiple Slice Component bundles, which servers in your Facility will have Secondary Cores installed, whether to install the OS Provisioning bundle, and so on. See Chapter 1, “SA Architecture” and SA Core Performance Scalability on page 41.

Administrative Tasks

Perform the pre-installation administration tasks, such as configuring your network. See Chapter 3, “Pre-Installation Requirements.”

Gather Environment Information

Gather information in preparation for the SA Installer interview. This includes such information as the name and ID of the Facility for the core, passwords, IP addresses, and so on.

You will use the Installer Interview Response File you created and saved during the installation of the First Core. See Chapter 5, “Prerequisites for the Installer Interview.”

IP Addresses

Verify that all Core Servers have unique IP addresses within the entire Multimaster Mesh.

Synchronize Time (UTC)

All servers in a Multimaster Mesh must use UTC. After you synchronize the time on all servers within a Facility, synchronize the time between the facilities in the Multimaster Mesh. Synchronize the time with an external time-server that uses Network Time Protocol (NTP) so that all servers are using the same Coordinated Universal Time (UTC).

Network Requirements

Verify that the Multimaster installation meets the same network requirements as a First Core installation, with the exceptions that each core must be on a different Local Area Network (LAN or VLAN). The cores must be in different broadcast domains. Exceptions are:

- If you run only one DHCP server and disable the DHCP servers on the other cores in the mesh.
- If you do not plan to do any OS provisioning, you can install several cores on a single network.
- If you are well-versed in DHCP configuration, you can configure your DHCP servers to handle several cores in the same network.

Subdomains

Ensure that each core in a Multimaster Mesh has a different subdomain so that managed servers can resolve the unqualified host names `spin`, `way`, and `theword`.

tsnnames.ora File

the `tnsnames.ora` file on the First Core contains entries for every Model Repository in the Multimaster Mesh. With this release, the `tnsnames.ora` file is automatically populated with the required entries for the Secondary Core being installed during the installation procedure.

For example entries, see tnsnames.ora File Requirements on page 206.

Oracle RDBMS Versions

Ensure that you do not have conflicting Oracle software versions within the Multimaster Mesh. See Multiple Oracle Versions and Multimaster Cores on page 188.

The Multimaster State Monitoring Utility

When installing a subsequent Secondary Core in an *existing, active Multimaster Mesh*, you must shut down the Data Access Engine and the Web Services Data Access Engine and then, on the server running the *Model Repository Multimaster Component* (part of the Infrastructure component), ensure that all transactions have been published and conflicts resolved before exporting Model Repository data.

See Prepare the Environment to Export First Core Model Repository Data on page 151.

In previous SA versions, this required inspecting the Model Repository Multimaster Component log files. SA now provides the Multimaster State monitoring utility to assist you in this task.

You must invoke the utility on the server that hosts the central Model Repository Multimaster Component (the Infrastructure Server).

Running the MSM Utility

To run the MSM utility, you must first set the environment:

```
export LD_LIBRARY_PATH=/opt/opsware/lib
```

Now you can enter the following to invoke the MSM utility:

```
cd /opt/opsware/spin/util
/opt/opsware/bin/python ./mm_state.pyc
```

The default for the MSM utility is to refresh the data display in near real time.

The MSM utility uses the Data Access Engine's library layer, therefore the Data Access Engine itself need not be running. However, the Model Repository and the Management and First Core gateways (if your Net10/Net11 traffic is tunneled) must be running.

Once the MSM utility is started, you will see a screen similar to this:

```
# Transactions Conflicting
From\To|   832   834 |
-------+-------------+
   832 |    --     0 |
   834 |     0    -- |
-------+-------------+
```

The screen above is the Transaction Conflict screen. It shows the source of the transaction for which a conflict has occurred in the left column and the destination in the top row.

If you press `h` at this screen, you will see the following options:

```
>>> Help:
  'a' for all counts
  'u' for unpublished counts
  'n' for not received counts
  'c' for conflict counts
  'e' for error counts
  'q' to exit

Press any key to continue
```

The MSM utility provides several monitoring options:

- `u` — show the count of transactions waiting to be published at each core.
- `n` — show the count of transactions published, but not received by the destination core.
- `c` — show the count of unresolved transaction conflicts at each core.
- `e` — show the count of all errors reading data from each core.
- `a` — show `u`, `n` and `c` data presented together. Note that, if the number of transaction is large, the column alignment may not be maintained.
- `q` — exit the MSM utility.

Select the optional views by pressing the associated key. Press `q` to exit.

Using the MSM Utility during Installation

To ensure that your system is quiesced as required, after shutting down the Data Access Engine and the Web Services Data Access Engine, invoke the MSM utility and monitor the outstanding transactions and unresolved conflicts. When these reach zero, then all transactions and conflicts are resolved and you can continue the installation.

Batch Mode

You can also invoke the MSM utility in batch mode using the `-b` command-line argument which will simply do a one time display of the current state and will not refresh the data.

```
export LD_LIBRARY_PATH=/opt/opsware/lib
cd /opt/opsware/spin/util
/opt/opsware/bin/python ./mm_state.pyc -b
```

Adding a Secondary Core to a Multimaster Mesh

This section describes how to add a subsequent Secondary SA Core to a Multimaster Mesh. There are several cross references, so ideally, you should scan this section first and make sure that you are prepared to perform all of the steps.

Throughout this section, the existing core that will act as the primary core for the Secondary core(s) you are adding is referred to as the First Core. The new core(s) that you are adding is called a Secondary or Subsequent Core.

When you add a Secondary Core to an existing Multimaster Mesh and you have applied a CORD patch to that mesh's existing core(s), (for example SA Cord release 7.80.01), your newly added Secondary Core will be version 7.80 and the patched Cores will be 7.80.01. You must apply the 7.80.01 Cord patch release to the newly added core to avoid a mixed version Mesh.

Overview of the Installation Process

The following are the typical phases of installing a Secondary Core:

1. *Pre-Installation*: Ensure that all installation prerequisites have been met, that you have the information needed to complete the Installer interview, that you have all necessary permissions to complete the installation, and that you have the SA installation DVDs. For more information, see Chapter 3, "Pre-Installation Requirements" and Chapter 5, "Prerequisites for the Installer Interview".

2 *Install the Oracle Database for the Secondary Core:* During this phase you will install the HP-supplied Oracle database for the Secondary Core(s) Model Repository. This database is automatically configured to work with the SA Model Repository. See Appendix A, “Oracle Setup for the Model Repository”

   Alternatively, you can install a database using the Oracle Universal Installer or use an existing Oracle 10g or 11g database installation (Oracle 9*i* is not supported) and skip Phase 2. However, there are database configuration requirements that must be met in order for such databases to be compatible with the SA Model Repository. See Appendix A, “Oracle Setup for the Model Repository”.

3 *Define a New Facility*: During this phase you will define the facility in which the new Secondary Core is to be installed.

4 *Export Model Repository Data*: During this phase, you will export the First Core's Model Repository data, copy the resulting export file (along with both versions of the cryptographic material bundle (the `db.e` and `tgz.e` files)) to the new Secondary Core server.

5 *Install Secondary Core*: During this phase, you will install the Secondary Core's components. You will also import the data exported in the previous phase into the database.

6 *Post Installation Tasks*: During this phase you must perform various post-installation tasks to complete the configuration of the new Secondary Core.

Before proceeding with the installation, ensure that you have also complied with the Prerequisites for Multimaster Mesh Installations on page 142.

Phase 1: Prepare for Installation

To prepare to add a subsequent Secondary Core to a Multimaster Mesh, perform the following tasks:

1 Locate the *SA Product Software DVD* and, if you will install the HP-supplied Oracle database for the Model Repository, the *Oracle_SA DVD*.

   See SA Installation Media on page 96, including the recommendation, “Copying the DVDs to a Local Disk.”

2 On the First Core’s Model Repository server and Infrastructure component server, and on each server of the new Secondary Core, mount the SA *Product Software* and *Oracle_SA* DVDs or NFS-mount the directory that contains a copy of the DVD contents.

The Installer must have *read/write root* access to the directories where it installs SA components, even on NFS-mounted network appliances.

3 On the Secondary Core Server on which you will install the Model Repository, open a terminal window and log in as root.

4 Change to the root directory:

```
cd /
```

Phase 2: Install the Oracle Database for the Model Repository on the Secondary Core

If you plan to use the HP-supplied Oracle database (for the Model repository), you must complete the tasks in this section to install the database. If you plan to use the Oracle Universal Installer to install an Oracle 10g or 11g database or you have an existing Oracle 10g or 11g database you plan to use for the Model Repository and have configured it as described in Appendix A, "Oracle Setup for the Model Repository", you can skip this section and go to Phase 3: Define the New Facility on page 148.

1 Mount the *Oracle_SAS DVD* and run the SA database Installer on each Secondary Cores Model Repository server, specifying the `--interview` argument.

You should use the response file you created when you installed the First Core for this Facility. Specify the full path to that response file, for example:

```
/opsware_system/opsware_installer/oracle_sas.sh -r
/usr/tmp/oiresponse.oracle_sas --interview
```

In the above example, both the `-r` (response file) and `--interview` options are invoked. This is because you need to use many of the parameter values you set for the First Core and specifying the response file created during the First Core installation will make those values the default during this interview. There may be, however, several new parameters for which you must supply initial values.

2 You will see the following screen:

Please select the interview mode. Simple mode uses default values for many of the configuration parameters. Advanced mode allows you to fully configure the installation.

```
1 - Simple Interview Mode
2 - Advanced Interview Mode

Please select the interview mode from the menu, type 'h' for help, 'q' to
quit: 1

The Installer will now interview you to obtain the installation parameters
it needs. You can use the following keys to navigate forward and backward
through the list of parameters:

Control-P - go to the previous parameter
Control-N - go to the next parameter
Return - accept the default (if any) and go to the next parameter
Control-F - finish parameter entry
Control-I - show this menu, plus information about the current parameter

Press Control-F when you are finished. The Opsware Installer will perform
a final validation check and write out a response file that will be used to
install the Opsware components.
```

Select `1` to go to the Simple Interview. In rare cases, the Advanced Interview is used to modify certain advanced parameters.

3 Complete the Interview. You are asked to supply or accept the default values for the following parameters. You can accept the defaults which are taken from the response file, `oiresponse.oracle_sas`:

- `truth.oaPwd` (`opsware_admin user`): This is the password used to connect to the Oracle database.

- `truth.servicename` (TNS name): the TNS name of the Model Repository instance for the Facility in which Installer is run.
- `truth.port`: The port on which the Model Repository database is listening. The default is 1521.

For example, if you choose the Simple Interview, you will see a screen similar to this:

```
Parameter 1 of 3 (truth.oaPwd)Please enter the password for the
opsware_admin user. This is the password used to connect to the Oracle
database.: opsware_admin
Validating... OK.

Parameter 2 of 3 (truth.servicename)Please enter the service name (aka TNS
name) of the Model Repository instance in the facility where Opsware
Installer is being run [truth]:
Validating... OK.

Parameter 3 of 3 (truth.port)Please enter the port on which the Model
Repository database is listening. [1521]:
Validating... OK.

All parameters have values.  Do you wish to finish the interview? (y/n): y


Concluding interview.

Interview complete.
```

4 The interview concludes. You see the following displayed:

```
Name of response file to write [/usr/tmp/oiresponse.oracle_sas]:
Response file written to /usr/tmp/oiresponse.oracle_sas.

Would you like to continue the installation using this response file? (y/
n): y
```

Press `y` You will then see this screen:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Oracle RDBMS for SAS
Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

Select `1` and press `c` to begin the database installation. When the installation is complete, a message to that effect is displayed and you will be returned to the command prompt. You can now continue with the Secondary Core installation, beginning with defining a new Facility for the new Secondary Core.

Phase 3: Define the New Facility

In this phase, you define a new Facility for the Secondary Core. After the interview in this phase, a new response file called `oiresponse.add_dc_to_mesh` is created.

The `oiresponse.add_dc_to_mesh` response file you create during this phase will be used later in Phase 4 when you export the Model Repository data, so save it in a secure location. `oiresponse.add_dc_to_mesh` is also used in SA upgrades so it is important that you store it in a location where you can find it again.

1 On the server that hosts the First Core's *Infrastructure Component bundle*, invoke the Installer with the `-r` (response file name) and the `--interview` (force interview mode) options.

For example:

```
/opsware_system/opsware_installer/install_opsware.sh -r
/usr/tmp/oiresponse.slices_master_typical --interview
```

In the above example, both the `-r` (response file) and `--interview` options are invoked. This is because you need to use many of the parameter values you set for the First Core and specifying the response file created during the First Core installation will make those values the default during this interview. There will be, however, several new parameters that you must supply initial values for.

The specified response file should be the one you used when you installed the current mesh's First Core components (in this example, `oiresponse.slices_master_typical`). You must specify the full path to the response file and to the installer script.

The Installer Installation Options screen displays the following:

```
Welcome to the Opsware Installer. Please select one of the following
installation options:

1 - Multimaster Opsware Core: First Core
2 - Multimaster Installation: Define New Facility; Export Model Repository
3 - Multimaster Installation: Subsequent Core
```

1 At the installation options prompt, select the second option:

```
2- Multimaster Installation: Define New Facility; Export Model Repository
```

2 At the interview mode prompt, select one of the following options:

```
1 - Simple Interview Mode
2 - Advanced Interview Mode
```

Choose Option 1 to use the default values for the configuration parameters.

Choose Option 2 to specify all configuration parameters including certain advanced parameters during the interview. For a description of the advanced parameters, see Chapter 5, Prerequisites for the Installer Interview, on page 77 of this guide.

3 In the simple interview, accept the interview prompt defaults. If there is no default, specify a value. Follow the on screen instructions to complete the interview. Since you specified the response file from the installation of the First Core, many of your responses will be displayed as the default during this installation. For more information about the installer prompts, see SA Installer Interview Prompts on page 78.

The installer displays default parameter values in square brackets [ ].

For the short name of this new Secondary Core (the `slaveTruth.dcNm` parameter), you must enter a new Facility name. This name must be unique within the Multimaster Mesh. That is, do not use the same Facility short name as the First Core or any other Secondary Core.

4 Complete the interview. When you have completed entering all of the required information, the Installer displays this message:

```
All parameters have values. Do you wish to finish the interview (y/n):
```

If you are satisfied with your answers, press y.

If you want to review or change your answers, press n. The installer displays the prompts again, showing in brackets [ ] the values that you just entered during the interview.

After modifying your responses, press y to finish the interview.

5 Save the response file. After completing the interview, the installer prompts you to provide a filename for the response file (the default is `oiresponse.add_dc_to_mesh`):

```
Name of response file to write
[/usr/tmp/oiresponse.add_dc_to_mesh]
```

The response file is a text file that contains the answers you entered during the interview. You can enter a path and name for the response file or accept the default location and name. In either case, write down the location and name of the response file for future reference.

The `oiresponse.add_dc_to_mesh` response file you create during this phase will be used later in Phase 4 when you export the Model Repository data, so save it in a secure location. `oiresponse.add_dc_to_mesh` is also used in SA upgrades so it is important that you store it in a location where you can find it again.

6 The Installer prompts you to indicate whether you want to continue the installation by using the current response file. Press `y` and go to Step 8 unless your component layout is like that discussed in the following note:

If the *First Core's Management Gateway* is on a different server than the *First Core's Model Repository*, enter `n`. Copy the response file to the First Core's *Management Gateway server* and go to **step 7**.

7 If you entered `n` in the previous step because the *First Core's Management Gateway* is on a different server than the First Core's Model Repository, log in to the server running the First Core's *Management Gateway* and invoke the Installer using the `-r` (specified response file) option. Be sure to specify the name and fully qualified path to the response file you created in **step 5**. For example:

```
/opsware_system/opsware_installer/install_opsware.sh -r
/usr/tmp/oiresponse.add_dc_to_mesh
```

8 You will see the following screen:

```
Welcome to the Opsware Installer. Please select one of the following
installation options:

1 - ( ) Define New Facility
```

At the Components prompt, select `Define New Facility` and press `c` to continue:

```
1 (*) Define New Facility
```

SA will begin defining the new facility. Wait for the installer to finish this operation before continuing with the Phase 4. During this process, the Installer registers the new Secondary Facility with the First Core's Model Repository, automatically generating a unique ID for the Facility.

If you are adding a third or more Secondary Cores to this mesh, before beginning Phase 4, you must allow enough time for the New Facility configuration you defined above to propagate to all other cores in the Multimaster Mesh. If this is the first or second Secondary Core installation, waiting for transactions to quiesce is not necessary.

Phase 4: Export the First Core Model Repository Data/Import Data into the New Secondary Core's Model Repository

Determine the Secondary Facility's unique ID that was assigned by SA in the previous phase.

You can find the Facility ID by:

1. Logging in to the SAS Web Client as the `admin` user in the First Core Facility.
2. From the Navigation panel, select **Environment ‰ Facilities**.
3. Select the link for the Secondary Facility. Note the Facility's ID.

Prepare the Environment to Export First Core Model Repository Data

1. On the server(s) that hosts the First Core's *Slice Component bundle(s)* (which contains the SA Command Center (OCC) and the Global File System), stop the Web Services Data Access Engine:

   ```
   /etc/init.d/opsware-sas stop twist
   ```

2. On the server that hosts the First Core's *Infrastructure Component bundle* (which contains the Data Access Engine), stop the Data Access Engine:

   ```
   /etc/init.d/opsware-sas stop spin
   ```

   If the First Core's *Command Center (OCC)* (Slice Component bundle) and the *Data Access Engine* (Infrastructure Component bundle) are installed on different servers, you must also run the preceding command on all the OCC server(s) hosts.

3. Stop the First Core's `vaultdaemon`:

   ```
   /etc/init.d/opsware-sas stop vaultdaemon
   ```

Before you begin the data export from the Model Repository, ensure that you do not have conflicting Oracle versions within the Multimaster Mesh. See Multiple Oracle Versions and Multimaster Cores on page 188.

Export the First Core Model Repository Data

1. Copy the response file `oiresponse.add_dc_to_mesh` that you created in Phase 3 from the First Core's *Infrastructure Component bundle* server to the server that hosts the First Core's *Model Repository.*
2. Log in as `root` to the server that hosts the First Core's *Model Repository* and invoke the installer with the `-r` (specify response file) option and specify the response file created by the last interview (in this example, `oiresponse.add_dc_to_mesh`). For example:

   ```
   /opsware_system/opsware_installer/install_opsware.sh -r
   /usr/tmp/oiresponse.add_dc_to_mesh
   ```

3. At the Components prompt, select the following option:

   ```
   2 (*) Export Model Repository
   ```

   Press `c` to continue. The installer exports the data from the Model Repository into a gzipped tar file called `truth_data.tar.gz`, which by default resides in the directory `/var/opt/opsware/truth` (or the directory that you specified as the `truth.dest` parameter value during the Installer interview).

Depending on the amount of data, the export can take 20 minutes or more. To track the progress of the export, open a new terminal window and run the following command where <*number*> is the most recent log file number plus one.

```
tail -f /var/log/opsware/install_opsware/truth/truth_exp<number>.log
```

Restart the First Core Components

1. On the server that hosts the First Core's *Infrastructure Component bundle* (which includes the Primary Data Access Engine) and on all cores where a *Slice Component bundle* is installed (which includes the Secondary Data Access Engine), start the Data Access Engines:

   ```
   /etc/init.d/opsware-sas start spin
   ```

   If the SA Command Center (OCC) and the Data Access Engine (Infrastructure Component bundle) are installed on different servers, you must also run the preceding command on the OCC (Slice Component bundle) server(s).

2. On the server(s) that host the First Core's *Slice component bundle(s)* (OCC and the Global File System Server), start the Web Services Data Access Engine:

   ```
   /etc/init.d/opsware-sas start twist
   ```

3. On the server that hosts the First Core's *Model Repository*, start the First Core's Model Repository Multimaster Component:

   ```
   /etc/init.d/opsware-sas start vaultdaemon
   ```

   Examine the logs for the Model Repository Multimaster Component to ensure that it started properly. These logs are located in the following directory:

   ```
   /var/log/opsware/vault
   ```

   The log files are named `vault.0.log`, `vault.1.log`, `vault.2.log`, and so on.

Copy the First Core Model Repository Export File to the New Facility

1. Copy the First Core's *Model Repository export file* (`truth_data.tar.gz`) to the server where you will install the *Secondary Core's Model Repository.*

The Unix `oracle` user must have read access to the `truth_data.tar.gz` file on the Secondary Core's Model Repository server.

Copy the Global Response File to the New Model Repository Server

1. Copy the *Global Response File* (`oiresponse.global`) from the First Core's *Model Repository server* to the new *Secondary Core's Model Repository server.*

On the First Core, the `oiresponse.global` file resides in the same directory as the Model Repository export file. The default directory is `/var/opt/opsware/truth`.

Copy the First Core Crypto Material to the New Model Repository

1. On all new *Secondary Core servers*, make the following directory:

   ```
   mkdir -p /var/opt/opsware/crypto/cadb/realm
   ```

2 Copy the cryptographic material database and Unix gzip tar file from the First Core's *Model Repository server* to every Secondary Core server. The cryptographic material database and Unix gzip tar file are located in:

```
/var/opt/opsware/crypto/cadb/realm/opsware-crypto.db.e
/var/opt/opsware/crypto/cadb/realm/opsware-crypto.tgz.e
```

You must copy these files to the same location on the new Secondary Core servers. Paths and filenames must match on all servers in the Multimaster Mesh.

The root user must have read access to these directories and files.

Phase 5: Install the New Secondary Core Components

1 Log in to the new *Secondary Core Model Repository* server and invoke the Installer using the `-r` argument (specified response file) and the `--interview` (force interview) argument. For this step, specify the fully qualified path to the `oiresponse.global` response file.

For example:

```
/opsware_system/opsware_installer/install_opsware.sh -r
/usr/tmp/oiresponse.global --interview
```

The Installer displays following options:

```
Welcome to the Opsware Installer. Please select one of the following
installation options:

1 - Multimaster Installation: First Core
2 - Multimaster Installation: Define New Facility; Export Model Repository
3 - Multimaster Installation: Subsequent Core
```

2 At the Installation Options prompt, select option 3 and press `c` to continue:

```
3- Multimaster Installation: Subsequent Core
```

3 At the Interview Mode prompt, select one of the following options:

```
1 - Simple Interview Mode
2 - Advanced Interview Mode
```

Choose Option 1 to use the default values for the configuration parameters.

Choose Option 2 to specify values for certain advanced parameters during the interview. These parameters rarely need to be changed. See Prerequisites for the Installer Interview on page 77 for more information about the advanced parameters.

4 If you chose the simple interview, accept the interview prompt defaults. Since you specified the global response file when invoking the installer, the parameters will display your default settings which you can accept.

The installer displays default parameter values in square brackets [ ].

Unless you have changed parameter values for the First Core since creating the Global Response File, accept the default values provided by that file. Parameter values in the Global Response File used to install the Secondary Core's components must match the values for the First Core.

Parameter values supplied during this interview must adhere to the following standards:

- The Facility ID (`truth.dcId`), Short Name (`truth.dcNm`), and Subdomain (`truth.dcSubDom`) must match the values generated when the Secondary Facility was defined in the First Core. You noted the Facility ID on page 151.
- The *Secondary Core's Authorization Domain* (`truth.authDom`) must match the value provided for the First Core.
- The path to the Model Repository data export file, `truth_data.tar.gz`, must be the same for both the *First Core's Model Repository serve*r and the *Secondary Core's Model Repository server.*
- The directories for the OS provisioning OS media must already exist on the *Secondary Core server* on which you will install the OS Provisioning Media Server component.

5 Complete the interview. When you have completed entering all of the required information, the Installer displays this message:

```
All parameters have values. Do you wish to finish the interview (y/n):
```

If you are satisfied with your answers, press y.

If you want to review or change your answers, press n. The installer displays the prompts again, showing in brackets [ ] the values that you previously entered.

After modifying your responses, press y to finish the interview.

6 Create the response file (the default file name is `oiresponse.slices_slave_typical`. After completing the interview, the installer prompts you to provide a filename for the response file:

```
Name of response file to write
[/usr/tmp/oiresponse.slices_slave_typical]
```

All of your interview responses will be written to a text response file and saved on the current server at the location you specify. You can enter the full path and name of the response file or accept the SA default location.

Record the fully qualified path to and name of the response file and store it where you can easily find it. You may need to use it again during future installations and upgrades.

7 The Installer prompts you to indicate whether you want to continue the installation by using the response file you just created. Select one of the following options:

- If you are satisfied with the responses you entered in the interview and you are ready to install the new Secondary Core's Model Repository now, enter `y` to continue and skip Step 8.
- If you do not want to install the Model Repository now, enter `n`. Step 8 shows how to restart the installation process.

8 *If you entered `y` in the previous step, skip this step*. If you entered `n` in the previous step, to restart the installation process, invoke the Installer with the `-r` option to specify the response file created by the interview. For example:

```
/opsware_system/opsware_installer/install_opsware.sh -r
/usr/tmp/oiresponse.slices_slave_typical
```

9 At the components prompt, select one or more components to install:

```
Welcome to the Opsware Installer.
Please select the components to install.

1 (*) Model Repository, Subsequent Core
2 (*) Infrastructure Components
3 (*) Slice
4 (*) OS Provisioning Component
```

```
Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

Single Host Component Installation: If you are installing all of the components of the Secondary Core on a single server, then you can press `a` to select all components and press `c` to continue. The SA Installer begins the installation of the selected SA Core Components.

Distributed Component Installation: If you plan to distribute components on multiple servers, you must run the SA Installer (specifying the response file created in Phase 5, step 8 on page 154) on each server on which you will distribute the components. You must also install the components in the order that they are listed on the *Component Selection* screen (you must install the Model Repository before the Infrastructure Component bundle and the Infrastructure Component bundle before the Slice Component bundle, and so on). In addition, if you plan to install multiple Slice Component bundles on the same or different hosts, you must install all of them before installing the OS Provisioning components.

To install the components, perform the following tasks:

a. Copy the response file generated in Phase 5, step 8 on page 154 to all other servers on which you will install component for this Secondary Core.

b. On each server in the *new Secondary Core*, run the Installer with the `-r` (specified response file) option, as shown in **step 8**. Select and install the components shown in **step 9**.

If the *Model Repository* exists on a server with no other SA components installed on it, you must install a *Server Agent* on that server. See the *SA User Guide: Server Automation* for instructions.

When distributing Core Components across multiple servers, you can install instances of the following components on different servers:

- Infrastructure Component bundle (one per core)
- Slice Component bundle(s) (multiple per core) (as of SA 7.80 the Slice Component bundle also includes a Software Repository)
- OS Provisioning Media Server (typically one per core)
- OS Provisioning Boot Server (typically one per core)

10. Perform the post-installation tasks in the next section of this chapter.

Multimaster Mesh Post-Installation Tasks

After you have added a new core to a Multimaster Mesh, you must perform the tasks described in this section.

Associate Customers with the New Facility

Associate the appropriate customers with each new Facility so that servers managed at that Facility are associated with the correct customers accounts. For more information, see the Customer Account Administration section of the *SA Policy Setter Guide*.

Update Permissions for the New Facility

After you have added a new Facility to your Multimaster Mesh, your SA users will not yet have the required permissions to access the new Facility. You must assign the required permissions to the user groups. For more information, see the User Group and Setup section of the *SA Administration Guide*.

Verify Multimaster Transaction Traffic

To verify Multimaster transaction traffic with the target Facility, perform the following tasks:

1. Log in to the SAS Web Client as any user who belongs to the SA System Administrators group.
2. From the Navigation panel, click Multimaster Tools under Administration. The View window appears.
3. In the State View Window, note the color of the status box beside each transaction.

   A *transaction* is a unit of change to a Model Repository database that consists of one or more updates to rows and has a globally unique transaction ID. If the transactions within the secondary Facility are green, the new SA Core is integrated into the Multimaster Mesh.

It is normal for some transactions to display an orange status (not sent) for a short period.

4. Click **Refresh** to refresh the cached data until all transactions display green.

For more information, see the Multimaster Mesh Administration section in the *SA Administration Guide*.

9 Satellite Installation

This section provides an overview of Satellites and Satellite installation requirements as well as instructions for installing a Satellite and post-installation tasks.

Satellite Installation Basics

A Satellite installation can be a solution for remote sites that do not have a large enough number of potential Managed Servers to justify a full SA Core installation by allowing you to install only the necessary Core Components for the remote site to function as a Satellite.

If you unsure of what a Satellite is, see SA Satellites on page 26 for an introduction to SA Satellites.

Installation Summary

The following is an overview of the Satellite installation process. For detailed instructions, see Satellite Installation on page 167.

1. Locate and mount the *SA Satellite Base DVD* (optionally, the *Satellite Base Including OS Provisioning DVD*) or NFS-mount the directory that contains a copy of the DVD contents
2. Run the SA Installer in interview mode using the response file used to install the First Core. The interviewer prompts you for information about your Satellite server environment and saves the information in a new Satellite response file.
3. Run the SA Installer and select the Satellite Gateway from the list of components to install. The Installer launches the Gateway Installer.
4. Supply or verify the parameter value during the interview.
5. If necessary, re-run the Installer to install OS Provisioning components.

Satellite Installation Requirements

Before you install a Satellite, verify that you meet following requirements.

- If your Satellite must be able to perform OS Provisioning in a Multimaster Mesh with an SA 7.80 Core, due to changes in OS Provisioning, the Satellite must also be SA 7.80. OS Provisioning does not work in a mixed Core/Satellite version installation.
- If you plan to install an OS Provisioning Boot Server and Media Server in the Satellite, you must adhere to the requirements in OS Provisioning: DHCP Proxying on page 61.
- The required packages listed in Solaris and Linux Requirements for Core Servers on page 47 must be installed on the Satellite server.
- The First and Secondary core(s) that will provide core component services to the Satellite must be running and accessible.
- The Satellite server must have network connectivity to the server running the First Core's Management Gateway.
- You must be able to log in to the First Core SAS Web Client as a member of the *Administrators* (`admin`) group as well as a member of a group that has Manage Gateway permissions.
- You must have root access on the First Core Model Repository host so that you can export and copy the database of cryptographic material to the Satellite server.
- The Satellite server uses UTC, as described in Time and Locale Requirements on page 65. The Satellite server's system time must be synchronized with the Primary Core server.
- If you plan to locate the Software Repository Cache on a network storage device, the network storage configuration must allow root write access over NFS to the directories in which the Software Repository Cache will be installed.
- You must know how to edit files using the `vi` editor. By default, the Gateway Installer launches the `vi` editor during the installation process, which you will use to edit the Gateway Properties File.

Required Open Ports

The ports listed in Table 40 must be open for use by the Satellite's Gateway. The port numbers listed in the table are default values. You can select other values during the installation.

Table 40 Open Ports for a Satellite Gateway

Port	Property Name in Gateway Properties File	Description
2001	`opswgw.TunnelDst`	The port used by a tunnel end-point listener. This port is used when you install other Gateways that tunnel to the Satellite Gateway on this Satellite.
3001	`opswgw.ProxyPort`	The proxy port on which Agents contact the Satellite Gateway.
4040	`opswgw.IdentPort`	The Gateway `ident` service port, used by the Software Repository Cache.

If you plan to install the OS Provisioning Boot Server and Media Server in the Satellite, then additional ports must be open. For a list of these ports, see Table 19 on page 59.

Required Entries in /etc/hosts

The Satellite's Software Repository Cache requires that the server hosting the cache have the following entries in the `/etc/hosts` file:

```
127.0.0.1  theword
127.0.0.1  wordcache
```

Required Packages for SUSE Linux Enterprise Server 9

In addition to the packages listed in Solaris and Linux Requirements for Core Servers on page 47, a Satellite on a server running SUSE Linux Enterprise Server 9 requires that the `compat-2004.7.1-1` package be installed.

Satellite Gateway Configuration

This section presents several Satellite topologies and the appropriate parameter values in the Gateway Properties File for those topologies. In the diagrams in this section, the arrows between Gateways represent tunneled connections. (A tunnel is a TCP connection between two Gateways that carries multiplexed TCP or UDP connections.) The servers labelled with the letter "A" represent Managed Servers that have Server Management Agents installed.

A Satellite Installation with a Single Core

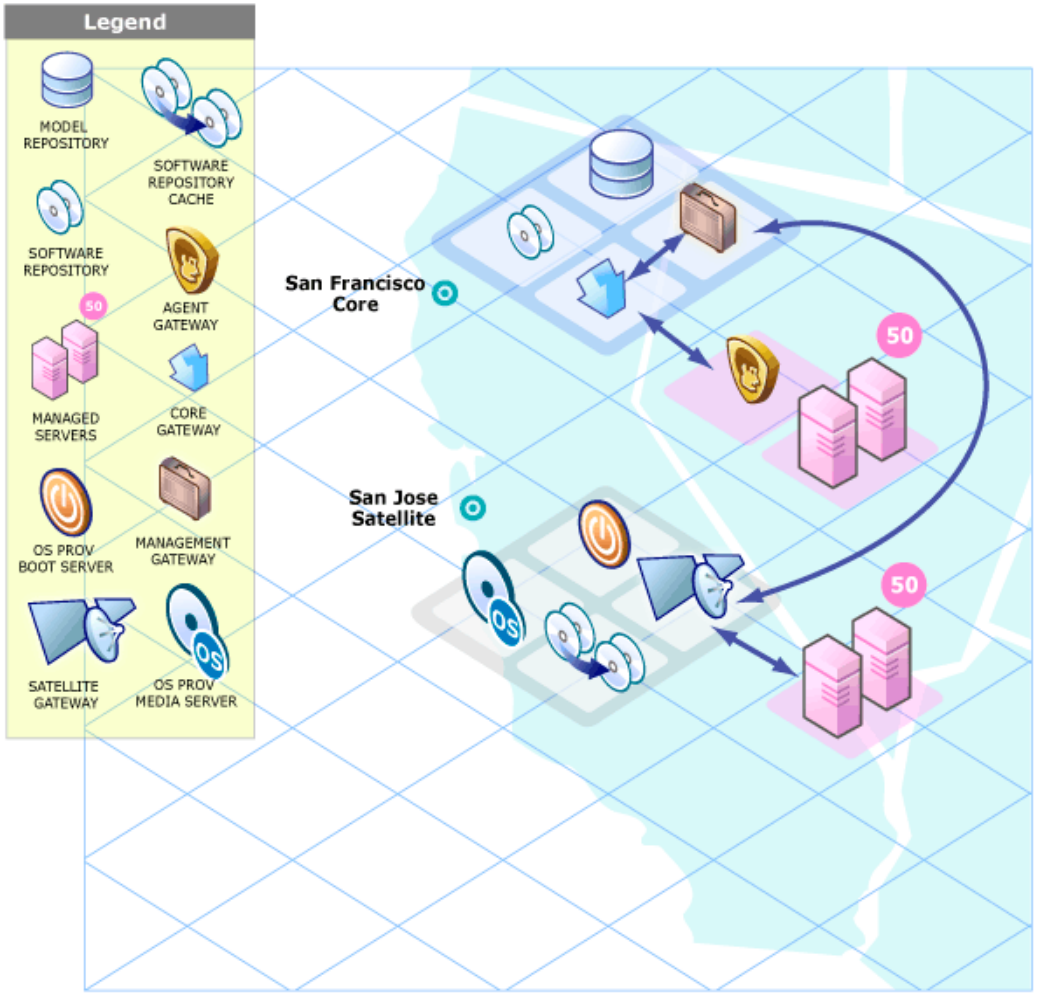
Figure 15 shows a single Satellite that has a tunneled connection to a Single Core's Management Gateway. In this example, the main facility is in San Francisco, and a smaller remote Satellite facility is in San Jose.

Server Management Agents running on the managed servers in the main San Francisco facility communicate with the San Francisco Core through an Agent Gateway. The Agent Gateway routes the requests to the Core Gateway which then communicates with the required Core Component(s).

The Server Management Agents on the managed servers in the San Jose facility connect to the San Francisco Core via tunneled TCP connections between the San Jose Satellite Gateway and the San Francisco Core's Management Gateway which, in turn, communicates with the San Francisco Core Gateway which ultimately communicates with the required Core Component(s).

A Satellite installation requires only the Software Repository Cache and Satellite Gateway components. The Software Repository Cache contains local copies of software packages that will be installed on the Satellite's managed servers. The Satellite Gateway multiplexes connections into and out of the Satellite via one or more tunnels to the Core's Management Gateway.

Figure 15 Single Satellite with a Single Core



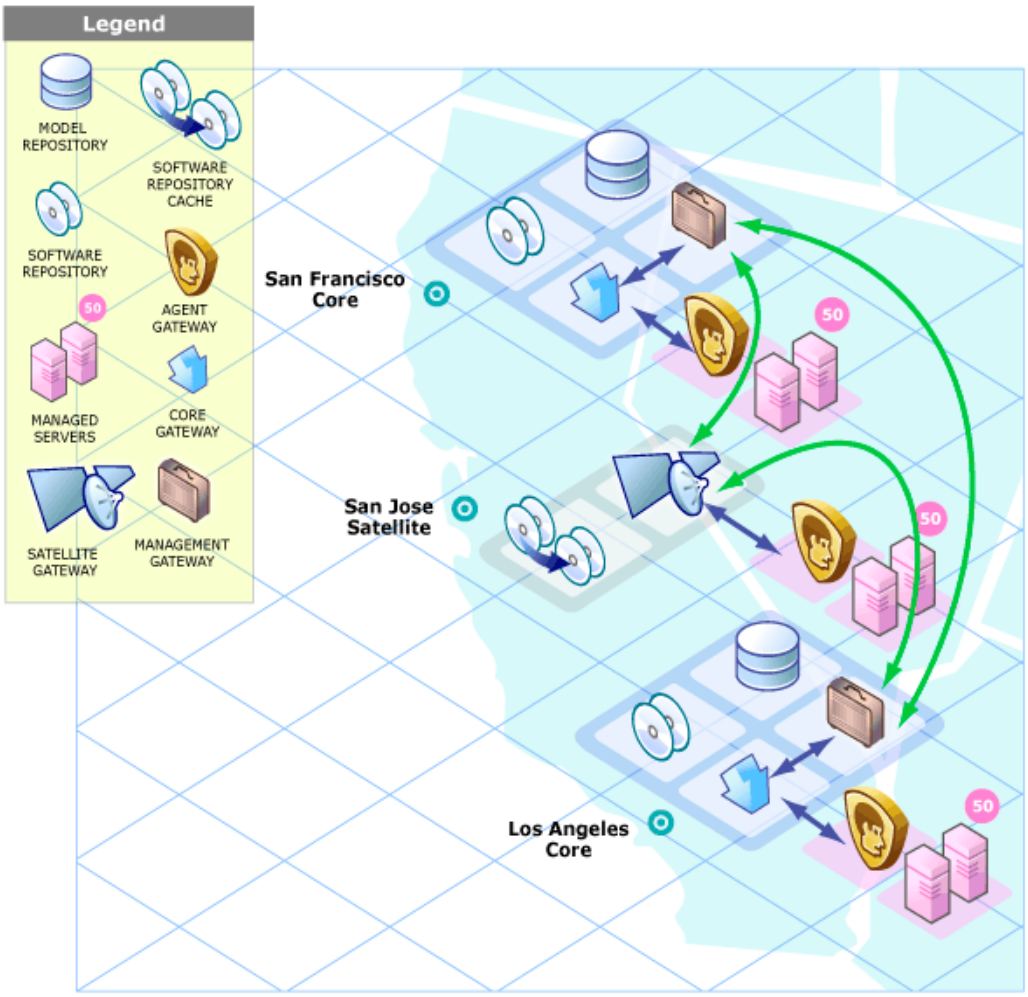
Satellite in a Multimaster Mesh

Figure 16 shows two Cores, San Francisco and Los Angeles, in a Multimaster Mesh. The Multimaster traffic passes through the Management Gateways. The Satellite Gateway in San Jose can route to either the San Francisco or Los Angeles Management Gateways, however the San Francisco Management Gateway is the primary route, the San Jose Management Gateway is a backup in case the San Francisco Management Gateway communications fail.

For the purposes of this example, assume that the communication link between the San Jose and San Francisco facilities is the fastest and has the most bandwidth. Therefore, during normal operations, the servers in San Jose are managed by the San Francisco Core.

Now, assume that the connection between San Jose and San Francisco has failed. The Satellite Gateway in San Jose can immediately begin to route communications through the Management Gateway in Los Angeles. (See Configuring Routing (Cost) on page 161.) allowing continued management of the San Jose servers.

Figure 16 Single Satellite in a Multimaster Mesh



The Gateway Properties File excerpt below would be appropriate for the San Jose Satellite Gateway. The first `opswgw.TunnelSrc` parameter points to the San Francisco Management Gateway; the second points to the Los Angeles Management Gateway. Both Management Gateways use the default port (2001) to listen for connection requests.

```
opswgw.Gateway=SanJose
opswgw.Realm=SanJose
opswgw.TunnelSrc=sanfran.myops.com:2001:100:0:/var/opt/opsware/crypto/
SanJose/opswgw.pem
opswgw.TunnelSrc=losang.myops.com:2001:200:0:/var/opt/opsware/crypto/SanJose/
opswgw.pem
```

Configuring Routing (Cost)

A Satellite Gateway routes traffic to only one Core Management Gateway at a time. The Management Gateway chooses the route with the lowest cost based on the third entry of the `opswgw.TunnelSrc` parameter.

In the Gateway Properties File excerpt above, the `opswgw.TunnelSrc` parameter entries specify that the cost from San Jose to San Francisco is 100 and the cost between San Jose and Los Angeles is 200. Therefore, the Satellite Gateway uses the San Francisco route, unless for some reason that connection becomes unavailable.

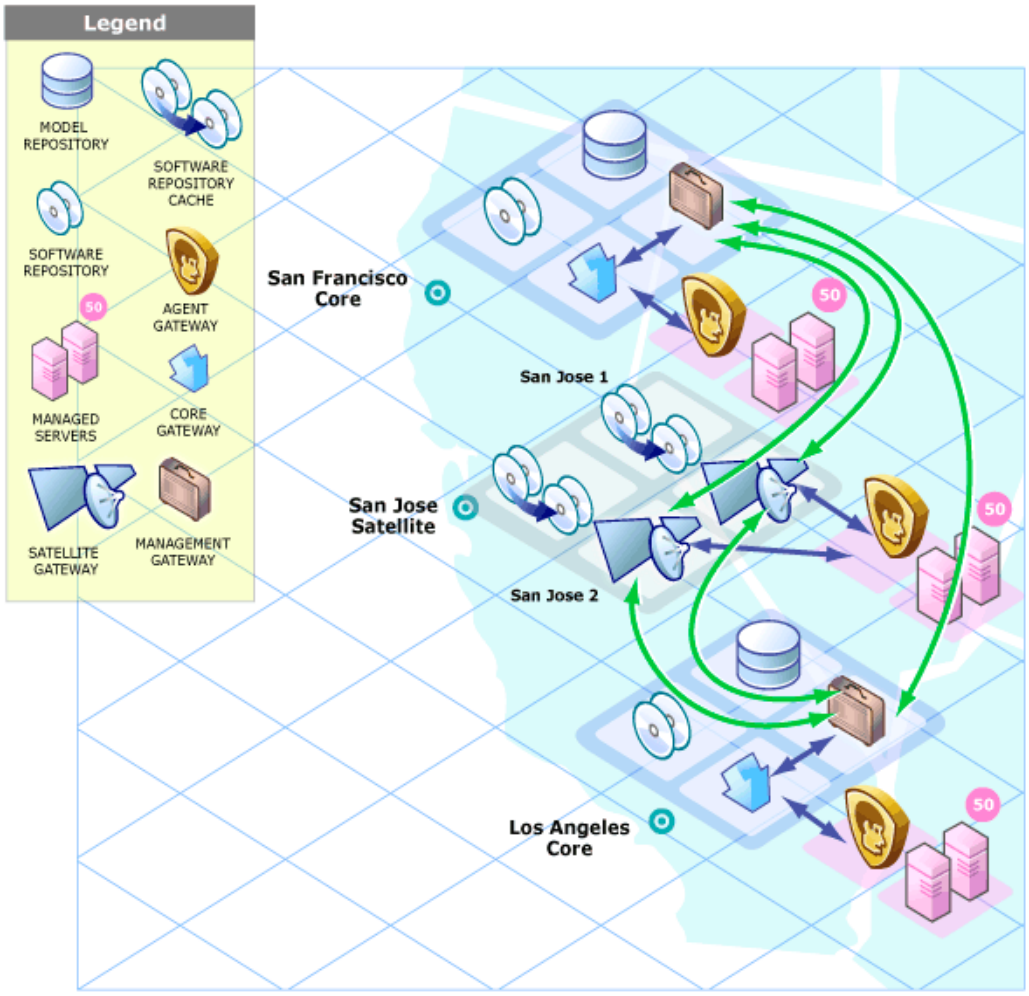
Multiple Gateways in a Satellite

The topology shown in Figure 17 provides failover capability in two ways. First, each Satellite Gateway in the San Jose facility tunnels to both the Los Angeles and San Francisco Management Gateways. If one of those Cores becomes unavailable, the other Core can take over management of the servers in San Jose.

Second, the Satellite Agents in San Jose point to both local Satellite Gateways (Gateway, San Jose 1 and gateway, San Jose 2). If one these gateways becomes unavailable, the Agents on the managed servers can communicate with a Management Gateway via the other Satellite's Gateway.

In this example, both Satellite Gateways in San Jose must belong to the same Realm. A Server Agent can communicate with any Gateway in the same Realm.

Figure 17 Multiple Gateways in a Satellite



The Gateway Properties File excerpt below would be appropriate the San Francisco Management Gateway:

```
opswgw.Gateway=cgw0-SanFrancisco
opswgw.Realm=SanFrancisco
opswgw.TunnelDst=2001:/var/opt/opsware/crypto/cgw0-SanFrancisco/opswgw.pem
```

The Management Gateway Properties File for the Los Angeles facility would have similar entries:

```
opswgw.Gateway=cgw0-LosAngeles
opswgw.Realm=LosAngeles
opswgw.TunnelDst=2001:/var/opt/opsware/crypto/cgw0-LosAngeles/opswgw.pem
opswgw.TunnelSrc=sanfran.myops.com:2001:1:0:/var/opt/opsware/crypto/
cgw0-LosAngeles/opswgw.pem
```

The Gateway Properties File excerpt below would be appropriate for the first Satellite Gateway in San Jose:

```
opswgw.Gateway=SanJose1
opswgw.Realm=SanJose
opswgw.TunnelSrc=sanfran.myops.com:2001:100:0:/var/opt/opsware/crypto/
SanJose1/opswgw.pem
opswgw.TunnelSrc=losang.myops.com:2001:200:0:/var/opt/opsware/crypto/
SanJose1/opswgw.pem
```

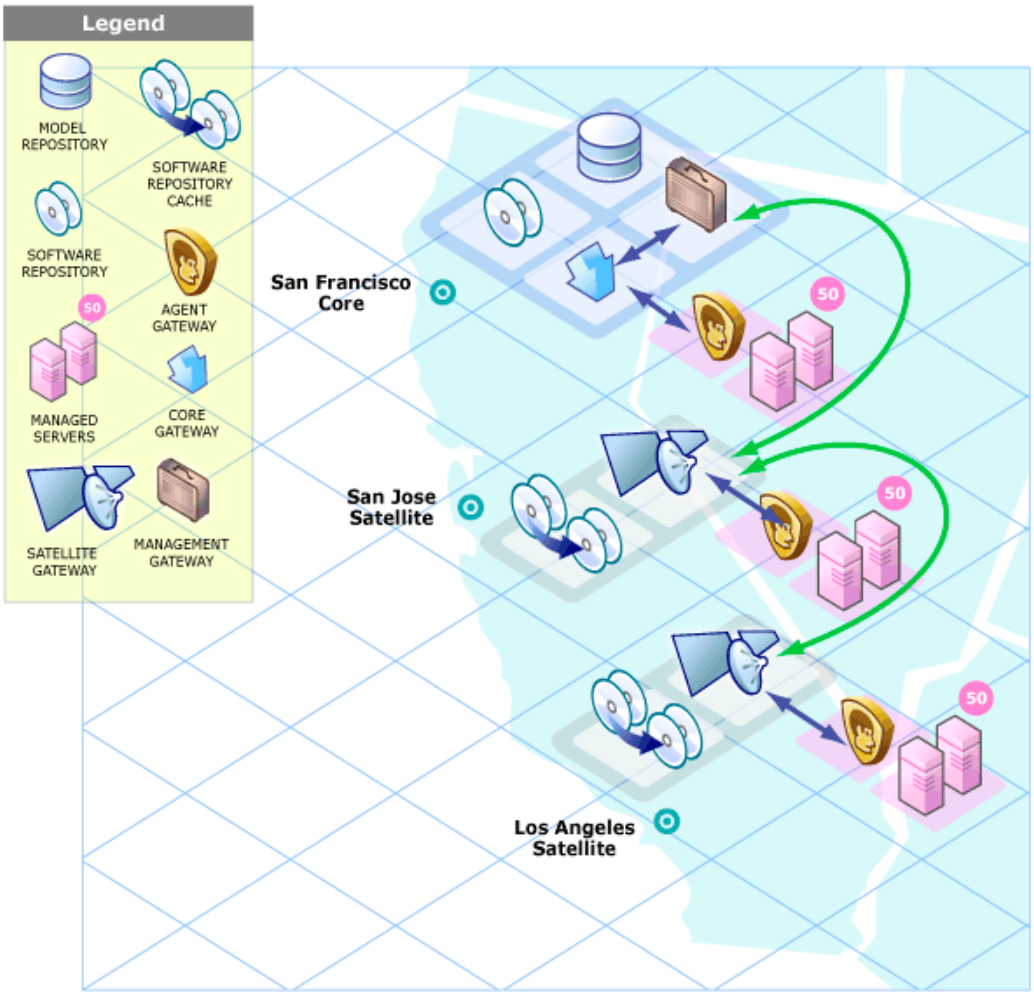
The Gateway Properties File excerpt below would be appropriate for the second Satellite Gateway in San Jose:

```
opswgw.Gateway=SanJose2
opswgw.Realm=SanJose
opswgw.TunnelSrc=sanfran.myops.com:2001:100:0:/var/opt/opsware/crypto/
SanJose2/opswgw.pem
opswgw.TunnelSrc=losang.myops.com:2001:200:0:/var/opt/opsware/crypto/
SanJose2/opswgw.pem
```

Cascading Satellites

Figure 18 is an example of cascading Satellites, a topology in which Satellite Gateways are connected to each other and a Core Management Gateway in a *chain* with the Core at the top of the chain. These Satellite Gateways must be in different Realms. (For more information, see “Managing the Software Repository Cache” in the *SA Administration Guide*.)

Figure 18 Cascading Satellites with a Standalone Core



The Gateway Properties File excerpt below would be appropriate for the San Francisco Management Gateway:

```
opswgw.Gateway=cgw0-SanFrancisco
opswgw.Realm=SanFrancisco
opswgw.TunnelDst=2001:/var/opt/opsware/crypto/cgw0-SanFrancisco/opswgw.pem
```

The Gateway Properties File excerpt below would be appropriate for the San Jose Satellite Gateway:

```
opswgw.Gateway=SanJose
opswgw.Realm=SanJose
opswgw.TunnelDst=2001:/var/opt/opsware/crypto/SanJose/opswgw.pem
opswgw.TunnelSrc=sanfran.myops.com:2001:100:0:/var/opt/opsware/crypto/
SanJose/opswgw.pem
```

The Gateway Properties File excerpt below would be appropriate for the Sunnyvale Satellite Gateway:

```
opswgw.Gateway=Sunnyvale
opswgw.Realm=Sunnyvale
opswgw.TunnelSrc=sanjose.myops.com:2001:100:256:/var/opt/opsware/crypto/
Sunnyvale/opswgw.pem
```

Limiting Bandwidth

In Figure 18, assume that the tunnel between Sunnyvale and San Jose shares a 512 kilobit/sec DSL connection with another application. Since this connection is relatively slow, you might want to limit the tunnel bandwidth to 256 kilobits/sec.

To limit the bandwidth, you would modify the Gateway Properties file and specify 256 for the fourth entry of the `opswgw.TunnelSrc` parameter. If you do not want to limit tunnel bandwidth, set this parameter to 0. Note that the bandwidth parameter is not used to determine the cost of a route. (See Configuring Routing (Cost) on page 161.)

Gateway Properties File

Each Gateway is configured by parameter values specified in a Gateway Properties File. The following sections describe some of the entries in a Satellite's Gateway Properties File that configure a Satellite Gateway for use with a Single Core.

opswgw.GWAddress

The `opswgw.GWAddress` parameter specifies the IP address of the server on which the Satellite Gateway is installed.

Facilities can belong to Realms. Realms are an SA concept that allow SA to manage servers on different networks without fear of IP address conflicts. A Realm is a logical entity that defines an IP namespace within which all managed server management IP addresses must be unique.

When a new Satellite Gateway is added to a Realm, the value of `opswgw.GWAddress` is dynamically added to the list of gateways that Agents in the same Realm can communicate with.

Although it is recommended that you use the IP address for `opswgw.GWAddress` you can use the hostname, however, if you do use the hostname then the value of the `opsw_gw_addr_list` parameter (used for Agent installations) must also use the hostname. If host names are used, they must be resolvable (with DNS or `/etc/hosts`) by the Agents that contact this gateway. Specifying IP addresses is recommended because it is less error prone.

opswgw.Realm

The `opswgw.Realm` parameter specifies the Realm to use for the Gateway. A Realm is a logical name for a group of IP addresses that can be contacted by a particular set of gateways. Realms enable SA to manage servers with overlapping IP addresses. (This situation can occur when the servers in a remote facility are behind NAT devices or firewalls.) The Realm plus the IP address uniquely identifies a managed server. Servers with overlapping IP addresses must reside in separate Realms.

opswgw.TunnelSrc

The `opswgw.TunnelSrc` parameter has five entries. The first two entries identify the remote host (`sanfran.myops.com`) and port (2001) where the Core Gateway listens for connections. Note that the host and port of the Satellite's `opswgw.TunnelSrc` parameter must match those of the Core's `opswgw.TunnelDst` parameter.

The next two entries specify the cost and bandwidth of the tunnel. (See Configuring Routing (Cost) on page 161 and Limiting Bandwidth on page 165.)

The last entry (`.../opswgw.pem`) is a certificate file in the Privacy Enhanced Mail (PEM) format. If you specify a certificate file, the data transmitted through the tunnel will be encrypted using SSL. The header of the certificate file includes the cipher choice and authentication options.

opswgw.DoNotRouteService and opswgw.HijackService

The parameters `opswgw.DoNotRouteService` and `opswgw.HijackService` must be enabled for this Satellite Gateway because the Satellite includes a Software Repository Cache. With these parameters enabled, when a Server Management Agent receives a request to access the Software Repository, the Satellite Gateway routes the request to the local Software Repository Cache rather than a remote cache. These parameters are disabled when commented out.

opswgw.ProxyPort and opswgw.IdentPort

The `opswgw.ProxyPort` parameter identifies the Satellite port through which Server Management Agents contact the Satellite Gateway. The `opswgw.IdentPort` parameter is used by an identity service required by the Software Repository Cache.

The following Gateway Property File excerpt shows some of the entries that would be appropriate for the San Jose Satellite in the example topology shown in Figure 15.

```
opswgw.Gateway=SanJose
opswgw.Realm=SanJose
opswgw.GWAddress=192.168.198.92
opswgw.TunnelSrc=sanfran.myops.com:2001:10:0:/var/opt/opsware/crypto/SanJose/
opswgw.pem
opswgw.DoNotRouteService=theword:1003
opswgw.DoNotRouteService=127.0.0.1:1003
opswgw.HijackService=wordcache:1003
opswgw.ProxyPort=3001
opswgw.IdentPort=4040

(Although the opswgw.TunnelSrc entry wraps around to the next line in this
listing, in the actual properties file, the entry is on a single line.)
```

The following Gateway Property File excerpt shows some of the entries that would be appropriate for the San Francisco facility's Core Gateway Properties File:

```
opswgw.Gateway=cgw0-SanFrancisco
opswgw.Realm=SanFrancisco

opswgw.TunnelDst=2001:/var/opt/opsware/crypto/cgw0-SanFrancisco/opswgw.pem
```

Satellite Installation

This section describes how to install a Satellite with the simple topology shown in Figure 15: a Satellite with a Single Core.

This topology has the following characteristics:

- The Satellite contains one Satellite Gateway and one Software Repository Cache, installed on the same server.
- The Satellite Gateway communicates with a single Management Gateway on the First Core server. No other gateways communicate with the Satellite Gateway.

Required Information

You will be prompted to supply the following information during the installation process. You can note the values in the right-hand column of Table 41:

Table 41 Satellite Installation Required Information

Requirement	Description	Value
Decryption password	The password required to decrypt cryptographic material	
First Core's Management Gateway IP	The IP address of the server running the First Core's Management Gateway	
Satellite Gateway server IP	The IP address of the server on which you will install the Satellite Gateway	
`opswgw.TunnelDst` parameter	The port number through which tunnel connections to the First Core's Management Gateway will pass. (The default port is 2001.) The Management Gateway listens on this port for connection requests from the Satellite Gateway. In the Management Gateway Properties File, this port specified with the `opswgw.TunnelDst` parameter The path to the Core's Gateway Properties file is: `/etc/opt/opsware/opswgw-mgw0-<facility>/opswgw.properties`	

Table 41 Satellite Installation Required Information (cont'd)

Requirement	Description	Value
`admin` user username and password	The `admin` username and password. You can also use the username and password of any user that belongs to the Administrators group	
Satellite Gateway name	The name of the new Satellite Gateway. The default directory on the Satellite server in which this Gateway will be installed is: `/opt/opsware/opswgw/bin`	
Realm name	The name of the new Realm to be serviced by the Satellite Gateway. SA uses the Realm name and the IP address of a managed server to uniquely identify a managed server. The Gateway Installer assigns the Realm name to the new Satellite facility. The Core and Satellite facility names must be different.	

You may want to name the Realm according to the physical location of the Satellite's data center, for example, the building, corporate site, or city. The SAS Web Client lists the facility names of the core and its Satellites.

Before Installing the New Satellite

- If you already have an SA Server Agent installed on the server you plan to use for the new Satellite, you must *uninstall* it before running the Satellite Installer.
- Make note that after the installation process completes, the new Satellite server is owned by the customer "`Opsware`". You should take into account any effects this may have on access rights before beginning the installation.

Phase 1: Prepare for Installation

1. Locate the *SA Satellite Base DVD* or, optionally, the SA *Satellite Base Including OS Provisioning DVD*.

   See SA Installation Media on page 96, including the recommendation, "Copying the DVDs to a Local Disk."

2. On the server where you will install the new Satellite, mount the Satellite Base DVD (optionally, the Satellite Base Including OS Provisioning DVD) or NFS-mount the directory that contains a copy of the DVD contents.

Whether you choose to install the "Satellite Base" DVD or the "Satellite Base Including OS Provisioning" DVD depends on whether you plan to install the OS Provisioning components. See SA Installer Interview Prompts on page 78 for information about each of the SA DVDs.

The Installer must have *read/write root* access to the directories where it will install the SA Core Components, including NFS-mounted network appliances.

3 In a terminal window, log in as root.

4 Create the Realm directory:

```
mkdir -p /var/opt/opsware/crypto/cadb/realm
```

5 Copy the database of cryptographic material and the gzipped tar file from any Core server in the facility to the Satellite server. On the Core server, the database and the gzipped tar file are located in:

```
/var/opt/opsware/crypto/cadb/realm/opsware-crypto.db.e
/var/opt/opsware/crypto/cadb/realm/opsware-crypto.tgz.e
```

The database of cryptographic material and the gzipped tar file must be copied to the same directory path and filenames on the Satellite server. The directory, database, and gzipped tar file must be readable by the root user.

6 Change to the root directory:

```
cd /
```

7 Go to Phase 2.

Phase 2: Complete the Installer Interview/Save the Response File

1 On the Satellite host, run the Installer script in interview mode by invoking it with no command-line options:

```
# /opsware_system/opsware_installer/install_opsware.sh
```

You must specify the full path to the script. The directory path shown in this step assumes that you copied an SA Satellite DVD (the Satellite Base DVD or the Satellite Base Including OS Provisioning DVD) to a local disk or a network share using the required directory structure.

2 At the Interview mode prompt, select one of the following options:

```
1 - Simple Interview Mode
2 - Advanced Interview Mode
```

Choose Option 1 to use the default values for certain configuration parameters.

Choose Option 2 to specify all configuration parameters during the interview.

3 Provide values for parameters presented during the interview or accept defaults.

During the interview, the value for the `mgw_address` parameter refers to the First Core's Management Gateway IP address, not the Satellite Gateway address.

For more information on the Installer prompts, see Chapter 5, "Prerequisites for the Installer Interview".

The parameter values requested during the interview are:

a `(truth.oaPwd)Please enter the password for the opsware_admin user`

b `(decrypt_passwd)Please enter the password for the cryptographic material`

c `(word_root)Please enter the root directory for the Package Repository [/var/opt/opsware/word]`

d `(media_server.linux_media)Please enter the pathname of the Linux media [/media/opsware/linux]`

e `(media_server.sunos_media)Please enter the pathname of the Solaris media [/media/opsware/sunos]`

f `(media_server.windows_media)Please enter the pathname of the Windows media [/media/opsware/windows]`

g `(boot_server.speed_duplex)Please enter the default network speed/duplex setting for Solaris SPARC servers [autoneg]`

h `(bootagent.host)Please enter the OS Provisioning Boot Server ip or hostname`

i `(agent_gw_list_args)Please enter the IP address and port number (ip:port) on which agents can contact the gateway in this facility`

j `(mgw_address) Please enter the IP address of the First Core's Management Gateway`

4 Supply values for the parameters. When you have completed entering all of the required information, the Installer displays this message:

```
All parameters have values. Do you wish to finish the interview (y/n):
```

If you are satisfied with your answers, press y.

If you want to review or change your answers, press n. The installer displays the prompts again, showing in brackets [ ] the values that you just entered during the interview.

After modifying your responses, press y to finish the interview.

5 Save the response file. After completing the interview, the installer prompts you to provide a filename for the response file:

```
Name of response file to write
[/usr/tmp/oiresponse.satellite]
```

The response file is a text file that contains the answers you entered during the interview. You can enter a path and name for the response file or accept the default location and name. In either case, write down the location and name of the response file for future reference.

6 The Installer prompts you to indicate whether you want to continue the installation by using the current response file. Select one of the following options:

- If you are satisfied with the responses you entered in the interview and you are ready to install the Satellite now, enter `y` to continue. Go to Phase 3
- If you do not want to install the Satellite now, enter `n`. Go to **step 7**.

7 *If you entered `y` in the previous step, skip this step and go to Phase 3.* If you entered `n` in the previous step, when you are ready to complete the installation later, log in to the server on which you will install the Satellite Gateway and invoke the Installer using the `-r` (response file) option and follow the instructions in Phase 3. The response file is the file you create in Phase 2, step 5 on page 170 (default: `/usr/tmp/oiresponse.satellite`). For example:

```
/opsware_system/opsware_installer/install_opsware.sh -r
<full_path_to_response_file>
```

Phase 3: Install the Satellite Gateway

1 The `Components to Install` menu is displayed:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Satellite Gateway (Interactive Install)
2 ( ) Software Repository Cache (wordcache)
3 ( ) OS Provisioning Boot Server
4 ( ) OS Provisioning Media Server
Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection: 1
```

At the components prompt, select Satellite Gateway (Interactive Install) to install the Satellite Gateway. Press `c` to continue.

The selections for the OS Provisioning Boot Server and OS Provisioning Media Server only appear if you are running the installation from the *Satellite Base Including OS Provisioning DVD*.

2 The Installer launches the Gateway Installer, which displays the following banner:

```
************************************************************
*                                                          *
*                Opsware Gateway Installer                 *
*          Copyright (C) 2004-2007: Opsware Inc.           *
*                    support@opsware.com                   *
*                                                          *
************************************************************
```

3 You should already have the required information for this step as described in the section Required Information on page 167. If not, get it now before continuing. The Gateway Installer displays the following message:

```
For a new install please have the following information
available before you begin:

1) Opsware administrator username and password.
2) The Realm name this Gateway will service.
3) If the Realm is new what type will it be.
4) The unique Gateway name for this Gateway.

Are you ready to proceed? [y/n]
```

Press `y`.

4 The Gateway Installer displays the following:

```
================================================
ISM install
================================================
. . .
```

5 Enter the name of the Realm for the Satellite Gateway you are installing:

```
================================================
Create/Verify Realm
================================================
```

```
Enter the Gateway's Realm name:
You entered '<realm-name>', is this correct [y/n]
```

6 The installer displays the following:

```
I must now contact an Opsware Core to continue the installation...
There are three ways this can be done:
 1) Via an existing Gateway's ProxyPort
 2) Via direct connections (no NATs)
 3) Via a temporary (local) Gateway
Enter option number: 3
```

There are three ways for the installer to contact the Core. A temporary (local) gateway is appropriate for almost all systems. Options 1 and 2 are available for rare network topologies. If you think you need to use Option 1 or 2, contact your support representative.

7 Enter the IP address of the server running the First Core Management Gateway at the following prompt:

```
Enter IP of a remote GW:
```

8 Enter the tunnel destination port for the Management Gateway at the following prompt. The default port is 2001.

```
Enter TunnelDst port of the remote GW: 2001
```

9 At the next prompt, press y.

```
Is the tunnel listener at <ip-addr:port>
using SSL? [y/n] y
```

10 Enter the `admin` username and password or the username and password of any SA user that belongs to the Administrators group:

```
================================================
Connect to Opsware
================================================

Log in to Opsware as an administrator

Enter username:admin
Enter password:
```

11 The Gateway Installer displays the following:

```
================================================
Checking time synchronization
================================================

Gateway time looks good.
```

12 At the next prompt, create a new Realm for this Satellite Gateway by selecting `Create a new Satellite DC named '<realm-name>'` and supplying the Realm name to create a new Satellite Gateway.

If you are adding the Realm to an existing DC, select `2` and supply the Realm name.

You also have the option to exit at this point by entering `3`.

```
================================================
Configure Realm
================================================

The realm '<realm-name>' does not exist.  You have two options:
  1) Create a new Satellite DC named '<realm-name>'.
```

```
  2) Add a new Realm, '<realm-name>', to an existing DC.
  3) Exit.
Enter option number: 1
```

13 At the next prompt, enter the name for the new Satellite Gateway that you are installing.

```
================================================
Gateway Configuration
================================================

Enter the Gateway's name:
```

14 The Gateway Installer opens the Gateway Properties File in the `vi` text editor. The following lines are at the top of this file:

```
#####################################################
#
# Opsware Gateway Properties File for a SAT Gateway
#
#####################################################
```

The fully qualified path to the Gateway Properties File is:

```
/etc/opt/opsware/<gateway_name>/opswgw.properties
```

Where `<gateway_name>` is the name you specified in **step 13**.

15 For the `opswgw.GWAddress` parameter, enter the IP address of the server on which you are installing this Satellite Gateway (the server you are running this installation on). For example:

```
opswgw.GWAddress=192.168.198.92
```

16 For the `opswgw.TunnelSrc` parameter, change the placeholder IP address of 10.0.0.11 to the IP address of the server running the Core Management Gateway. The port following the IP address is the tunnel destination of the Core Gateway. (The default port is 2001.) For example:

```
opswgw.TunnelSrc=192.168.165.242:2001:100:0:/var/opt/opsware/crypto/
<gateway-name>/opswgw.pem
```

17 You will be installing a Software Repository Cache in a later step, so verify that the following lines in the Gateway Properties File are *not* commented out:

```
opswgw.DoNotRouteService=theword:1003
opswgw.DoNotRouteService=127.0.0.1:1003
opswgw.HijackService=wordcache:1003
```

These parameters are disabled when they are commented out. If the Gateway and Software Repository Cache are to reside on the same server, these parameters must be enabled by removing the commenting.

18 After you've finished editing the Gateway Properties File, save it and exit `vi`. You will see a prompt asking if you want to proceed. Enter `y`. The Gateway Installer performs several more tasks then displays the following messages:

```
Gateway Crypto Generation
. . .
Wordcache Crypto Generation
. . .
Starting Opsware Gateway
. . .
Verify Gateway Startup
```

When the installer is finished, it displays the following:

```
Opsware Gateway Installed!
```

Phase 4: Install the Software Repository Cache

You must install the components in the order they are listed. For example, you must install the Software Repository Cache before the OS Provisioning Boot Server.

1 Invoke the Installer again with the `-r` option to specify the response file created by the interview in step 5 on page 170:

```
/opsware_system/opsware_installer/install_opsware.sh -r
<full_path_to_response_file>
```

2 At the components prompt, select one or more components to install:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Software Repository Cache (wordcache)
2 ( ) OS Provisioning Boot Server
3 ( ) OS Provisioning Media Server

Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

Select `Software Repository Cache (wordcache)`. Press c to continue. The SA Installer installs the cache.

The Software Repository Cache is required and must be installed on the same server as the Satellite Gateway.

The selections for the OS Provisioning Boot Server and OS Provisioning Media Server only appear, if you are running the installation from the Satellite Base Including OS Provisioning DVD.

Phase 5: Install the OS Provisioning Components

The OS Provisioning *Boot Server* and *Media Server* are required only if you want to use the OS Provisioning feature in the Satellite. The OS Provisioning Boot Server and Media Server can reside on a different server than the Gateway and Software Repository Cache.

1 [**OS Provisioning Components on Satellite Host**] If you are installing the OS Provisioning components on a non-Satellite host, go to **step 3**.

If you are installing the OS Provisioning components on the same host as the Satellite Gateway and the Software Repository Cache, invoke the Installer again with the `-r` option to specify the response file created by the interview in step 5 on page 170:

```
/opsware_system/opsware_installer/install_opsware.sh -r
<full_path_to_response_file>
```

2 At the components prompt, select one or more components to install:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Software Repository Cache (wordcache)
```

```
2 ( ) OS Provisioning Boot Server
3 ( ) OS Provisioning Media Server

Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

Select `OS Provisioning Boot Server` and `OS Provisioning Media Server`. Press c to continue. the SA Installer installs the OS Provisioning components.

If you plan to install the OS provisioning Boot Server on the Satellite, but install the Media Server on a different host, select only `OS Provisioning Boot Server`, install that component then log on to the server that will host the Media Server and invoke the install script again with the response file specified and install the Media Server.

3 **[OS Provisioning Components on non-Satellite Host]** If you are installing the OS Provisioning components on a *different* server than the Satellite Gateway and Software Repository Cache, follow the instructions in this step.

a Copy the database of cryptographic material and the gzipped tar file from the *Satellite Gateway host* to the OS Provisioning components host. These file are found on the Satellite host in the following location:

```
/var/opt/opsware/crypto/cadb/realm/opsware-crypto.db.e
/var/opt/opsware/crypto/cadb/realm/opsware-crypto.tgz.e
```

The database of cryptographic material and the gzipped tar file must have the same paths and filenames on both servers. The directory and files also need to be readable by the root user.

b Copy the response file created by the interview in step 5 on page 170 to the server that will host the OS Provisioning components.

c Invoke the Installer again with the `-r` option to specify the response file created by the interview in step 5 on page 170:

```
/opsware_system/opsware_installer/install_opsware.sh -r
<full_path_to_response_file>
```

d At the components prompt, select one or more components to install:

```
Welcome to the Opsware Installer.

Please select the components to install.

1 ( ) Software Repository Cache (wordcache)
2 ( ) OS Provisioning Boot Server
3 ( ) OS Provisioning Media Server

Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

Select `OS Provisioning Boot Server` and `OS Provisioning Media Server`. Press c to continue. the SA Installer installs the OS Provisioning components.

Post-Satellite Installation Tasks

After you install the Satellite, perform the tasks listed in the following sections. For more information, see the *Satellite Administration* section of the *SA Administration Guide*.

Facility Permission Settings

The SA Gateway Installer assigns the Realm name to the facility name of the Satellite. To access managed servers in the Satellite, an SA user must belong to a group that has the necessary permissions for the Satellite's facility. Until you set the facility permissions, SA users cannot view or modify the managed servers associated with the Satellite's facility. For example, you might set the permissions for the Satellite facility to Read & Write for the Advanced Users group, enabling members of this group to modify the servers managed by the Satellite.

For instructions, see "Setting the Facility Permissions of a User Group" in the *SA Administration Guide*.

Checking the Satellite Gateway

To verify that the Core Management Gateway is communicating with the Satellite Gateway, perform the following steps:

1. Log in to the SAS Web Client as a member of a users group that has the Manage Gateway permission.
2. From the Navigation panel, click **Administration** ‰ **Gateway**.
3. Verify that the upper left corner of the Manage Gateway page displays a link for the new Satellite Gateway.

   If the Manage Gateway page does not display the link for the Satellite, you might need to correct the properties file of the Satellite Gateway. The full path name of the properties file follows:

   ```
   /etc/opt/opsware/opswgw-cgw0-<facility>/opswgw.properties
   ```

   If you modify the properties file, you must restart the Satellite Gateway:

   ```
   /etc/init.d/opsware-sas restart opswgw-cgw0
   ```
4. Log in to the SAS Web Client as a member of a users group that has the Read (or Read & Write) permission on the Satellite's facility.
5. From the Navigation panel, click **Servers** ‰ **Manage Servers**.
6. Verify that the Manage Server page displays the host name of the Satellite server.

Enabling the Display of Realm Information

By default, the SAS Web Client does not display realm information, which is needed by users who manage Gateways and Software Repository Caches.

To enable access to the realm information, perform the following steps:

1. Log into the SAS Web Client as a user that belongs the Administrators group and to a group that has the Configure Opsware permission.
2. From the navigation panel, click Administration ‰ System Configuration.
3. Select the Opsware Server Automation System Web Client link.

4 In the System Configuration page, for the name `owm.features.Realms.allow`, type the value `true`.

5 Click **Save**.

DHCP Configuration for OS Provisioning

After you install the OS Provisioning Boot Server component, you must set up a DHCP server. For more information, see DHCP Configuration for OS Provisioning on page 127.

10 SA Configuration

SA Configuration

After you have installed the first SA Core, whether as part of a Single Core or Multimaster installation, the SA Core Components will be running and you will be able to log in to that Core's SAS Web Client. You can now configure SA so that end users can start managing servers in their operational environment.

The following sections provide a general outline of the SA configuration tasks you will need to do and pointers to the HP documentation that contains the detailed instructions needed to complete the tasks.

Configure e-mail Alerts

You can configure SA to send e-mail alerts to the SA administrator (or other designated users) when certain conditions are met, such as Managed Server error conditions, Multimaster Mesh conflicts, and Code Deployment and Rollback errors. To do so, your e-mail administrator must configure the SA Core and Managed Servers as Sendmail clients. You should configure e-mail alerts In the SAS Web Client when you install Server Agents on your managed servers. For information about e-mail alerts, see the *SA Administration Guide*.

Set Up SA Groups and Users

You must assign the necessary access rights and permissions to SA administrators, users, and user groups. For example, to log in to the SAS Web Client, you specify a user name and password. Each user belongs to a user group, and each user group has a set of permissions that control access to features (actions), managed servers, and folders. For information about user access rights and permissions, see the *SA Administration Guide*.

Create SA Customers

When you installed the First Core, whether Single Core or Multimaster, you specified a single default SA customer. For information about creating and assigning additional customers to a facility, see the *SA Policy Setter Guide*.

Define Software Management Policies

Software policies allow you to install software and configure applications simultaneously. A software policy can contain packages, RPM packages, patches, application configurations, and other software policies. After creating a software policy, you can attach it to servers or groups of servers. When you remediate a server or group of servers, the patches, packages, RPM packages, and application configurations specified in the attached policy are automatically installed and applied.

See the *SA Policy Setter Guide* for information.

Deploy Server Agents on Unmanaged Servers

After you install an Server Agent on an unmanaged server, it can be managed by HP Server Automation. For more information about deploying Server Agents on your unmanaged servers, see the *SA User Guide: Server Automation*.

Prepare SA for OS Provisioning

OS Provisioning is a feature that allows you to remotely install and uninstall operating systems (and related configurations, packages, and applications) on your servers. During OS Provisioning, a Server Agent is also installed, allowing the server to be immediately managed. For more information about configuring OS Provisioning, see the *SA Policy Setter Guide*.

Prepare SA for Patch Management

The Patch Management for Windows feature enables you to identify, install, and remove Microsoft® Windows patches. With the SA Client user interface, you can identify and install patches for the Windows 2000, Windows 2003, and Windows NT4.0 operating systems. These patches include Service Packs, Update Rollups, and hotfixes. This feature also supports patching on 64 bit for Windows 2003 operating systems and for 32 bit for Windows XP operating systems.

For information about Windows patch management, see the *SA User's Guide: Application Automation*.

SA Monitoring

SA provides several methods that you can use to ensure that your system is performing correctly:

— **Agent reachability tests**: to determine the current reachability of a specific Agent, you can run a Communication Test in the SAS Web Client to find those servers that have unreachable agents. For more information about the Communications Test, see the *SA User Guide: Server Automation*.

— **System Diagnostic tests**: several system diagnostics tests are available in the SAS Web Client that can help you determine that your SA installation is operating correctly and help you troubleshoot when there are problems. For more information about the SA System Diagnostic Tests, see the *SA Administration Guide*.

— **Core Component logs**: SA components have logs that can help you troubleshoot problems. For more information about Component Logs, see the *SA Administration Guide*.

11 SA Core Uninstallation

This section describes how to uninstall a Single Core, remove a core from a Multimaster Mesh, and how to uninstall all cores of a Multimaster Mesh.

Uninstall Basics

There are several reasons that you might choose to uninstall an SA core:

- Removing test installations
- Removing demonstration installations
- Merging or modifying a Facility's Multimaster Mesh Cores
- Decomissioning or moving a Facility

Make backups of your Model Repository, Software repository, and your database of cryptographic material unless you are certain that you no longer need that data, because a complete Core uninstall also removes the Model Repository and the cryptographic material database and permanently deletes all the data.You can preserve the SA data in the Model Repository database by doing a database backup before uninstalling.

Before you uninstall an SA Core, you should back up the Oracle database running on the server where that core's Model Repository is installed. See Oracle Database Backup Methods on page 219.

Like an SA installation, the uninstall is done using a script that you run from the server hosting the Core to be uninstalled.

Procedures for Uninstalling Cores

You can perform any of the following four uninstallation procedures according to your requirements:

- Uninstall a Single Core
- Uninstall a Single Core in a Multimaster Mesh
- Uninstall All Cores in a Multimaster Mesh
  - Decommission a Facility using the SAS Web Client

Uninstall a Single Core

To uninstall a Single Core, perform the following tasks:

1. Before uninstalling a single core, you must deactivate all servers that host components for that Core using the SAS Web Client. For more information about deactivating Core Component servers, see *Deactivating a Server* in the *Basic Server Management Tasks* section of the *SA User Guide: Server Automation*.
2. On the server hosting the core to be uninstalled, log in as root.
3. Change to the root directory:

   ```
   cd /
   ```

4. Run the `uninstall_opsware.sh` script with the -r (specify response file) argument. You need to use the response file you created when you installed the SA Core you are uninstalling:

   ```
   /opsware_system/opsware_installer/uninstall_ opsware.sh -r <response-file>
   ```

   You must specify the full path the response file.

5. A menu similar to the following appears:

   ```
   Welcome to the Opsware Installer.
   Please select the components to uninstall.
   1 ( ) OS Provisioning
   2 ( ) Slice
   3 ( ) Infrastructure
   4 ( ) Model Repository
   5 ( ) Oracle RDBMS for SAS
   ```

   Select one or more or all components to uninstall:

   Press `a` to select `all` components. If you must uninstall components one-at-a-time, for example due to a custom installation, the components must be uninstalled in the order they appear on the menu above. For example, you would first log on to the OS Provisioning component host, run `uninstall_ opsware.sh -r <response-file>` and uninstall that component, then log into the Slice Component bundle host and run the uninstall script to remove that component, and so on down the list.

   You will be asked if you want to preserve the database of Cryptographic Material. If you respond y, the directory containing the database will not be removed during the uninstall.

   You will also see this prompt:

   ```
   Are you absolutely sure you want to remove users' OGFS home and audit
   directories? (home and audit directories will only be removed if they are
   stored on the Software Repository server) (y/n)?
   ```

Select y if you want to remove the OGFS home and audit directories. If you press n, the directories will not be removed. Note that, if you have placed the OGFS home and audit directories on a server other than the server hosting the Software Repository, the uninstall will not remove those directories even if you press y.

6 After you have uninstalled all components, remove the `/var/opt/opsware/install_opsware` directory.

If you specified during the uninstall that you want to preserve the database of cryptographic material, you should *not* delete the `/var/opt/opsware/crypto` directory. This directory contains the database of your cryptographic material.

Uninstall a Single Core in a Multimaster Mesh

Do not uninstall the First Core (Primary Core) unless you plan to uninstall the entire Multimaster Mesh and all its cores. See Uninstall All Cores in a Multimaster Mesh on page 185 in this chapter for more information.This section describes only uninstalling *Secondary Cores* from a Multimaster Mesh.

To uninstall a single secondary core in a Multimaster Mesh, perform the following tasks

1 Log in to any SAS Web Client available for that Mesh:

a Using the System Configuration feature, update the `listeners` configuration parameter by removing the entry for the secondary core that you will uninstall. Update the listeners parameter by selecting **Model Repository, Multimaster Component** on the **System Configuration** page.

b If the secondary core to be uninstalled has a Data Access Engine that is currently serving in the Primary (Multimaster central) role, you must first assign Data Access Engine in another Core to serve as the Primary Data Access Engine.

See *Reassigning the Data Access Engine to a Secondary Role* in the *SA Administration Guide*.

c Verify that all transactions have propagated to the other facilities in the Multimaster Mesh.

For more information about verifying transaction traffic, see Verify Multimaster Transaction Traffic on page 156.

2 Decommission the facility for the core you will uninstall.

a See “Decommission a Facility using the SAS Web Client” on page 186.

b On the *Infrastructure Component bundle host* in the core you are decommisioning, run the following command:

```
/opt/opsware/bin/python2
disk001/opsware_installer/tools/reload_vaults.pyc  --certfile
/var/opt/opsware/crypto/gateway/spin.srv
```

Successful output will be similar to this:

```
Core ID         Peers IDs Known To This Core
-------         --------------------------------
<nnn>            <nnn>
```

3 Stop and start the *Model Repository Multimaster Component* in all cores except the core that you will uninstall by entering the following command as root on the server running the engine:

```
/etc/init.d/opsware-sas stop vaultdaemon
```

```
/etc/init.d/opsware-sas start vaultdaemon
```

4 Stop the Command Center (OCC) component (part of the Slice Component bundle) in the core that you will uninstall by entering the following command as root:

```
/etc/init.d/opsware-sas stop occ.server
```

5 In the core that you will uninstall, stop all *Data Access Engines* (part of the Infrastructure Component bundle).

Log in as root to the server where the Data Access Engine is running and enter the following command:

```
/etc/init.d/opsware-sas stop spin
```

6 If the Command Center and the Data Access Engine are installed on different servers, you must also run the `spin stop` command on the Command Center server.

7 Stop the *Model Repository Multimaster Component* in the core that you will uninstall by entering the following command as root on the server running the engine:

```
/etc/init.d/opsware-sas stop vaultdaemon
```

8 Restart the *Data Access Engine* that is serving as the Primary Data Access Engine (Multimaster Central) by entering the following commands as root:

```
/etc/init.d/opsware-sas stop spin

/etc/init.d/opsware-sas start spin
```

9 On the server hosting the core to be uninstalled, log in as root.

10 Change to the root directory:

```
cd /
```

11 Run the `uninstall_opsware.sh` script:

```
/opsware_system/opsware_installer/uninstall_ opsware.sh -r <response-file>
```

12 At the components prompt, select one or more or all components to uninstall:

```
Welcome to the Opsware Installer.
Please select the components to uninstall.
1 ( ) OS Provisioning
2 ( ) Slice
3 ( ) Infrastructure
2 ( ) Model Repository
1 ( ) Oracle RDBMS for SAS
```

Select `a` for all. If you want to uninstall components separately, they must be uninstalled in the order they appear on the menu above. To do so, enter the number of the component to uninstall.

If the gateway does not run on a separate server, uninstall it last. You will be asked if you want to preserve the database of Cryptographic Material. If you respond y, the directory containing the database will not be removed during the uninstall.

You will also see this prompt:

```
Are you absolutely sure you want to remove users' OGFS home and audit
directories? (home and audit directories will only be removed if they are
stored on the Software Repository server) (y/n)?
```

Select y if you want to remove the OGFS home and audit directories. If you press n, the directories will not be removed. If you chose to place the OGFS home and audit directories on a server other than the server hosting the Software Repository, the uninstall will not remove those directories even if you press y.

If you installed the core using Custom Mode, it is important that you uninstall the components in the reverse order that they were installed.

13 After the uninstall has completed, remove the `/var/opt/opsware/install_opsware` directory.

If you specified during the uninstall that you want to preserve the database of cryptographic material, you should *not* delete the `/var/opt/opsware/crypto` directory. This directory contains the database of cryptographic material.

Uninstall All Cores in a Multimaster Mesh

To uninstall all cores in a Multimaster Mesh, perform the following steps:

1 Stop the OCC by logging on as root to the server where the OCC is running and enter the following command:

```
/etc/init.d/opsware-sas stop occ.server
```

2 Stop the Data Access Engine.

Log in as root to the server where the Data Access Engine is running and enter the following command:

```
/etc/init.d/opsware-sas stop spin
```

If the OCC and the Data Access Engine are installed on different servers, you must also run the stop spin command on the OCC server.

3 Stop the Model Repository Multimaster Component in all cores by logging in to the servers running the engines and running the following command as root:

```
/etc/init.d/opsware-sas stop vaultdaemon
```

4 In each core, uninstall the SA components on the servers where they are installed. On the servers hosting the cores to be uninstalled, log in as root.

5 Change to the root directory:

```
cd /
```

6 Run the `uninstall_opsware.sh` script:

```
/opsware_system/opsware_installer/uninstall_ opsware.sh -r <response-file>
```

7 At the components prompt, select one or more or all components to uninstall:

```
Welcome to the Opsware Installer.
Please select the components to uninstall.
1 ( ) OS Provisioning
2 ( ) Slice
3 ( ) Infrastructure
2 ( ) Model Repository
1 ( ) Oracle RDBMS for SAS
```

Select `a` for all. If you want to uninstall components separately, they must be uninstalled in the order they appear on the menu above. To do so, enter the number of the component to uninstall.

If the gateway does not run on a separate server, uninstall it last. You will be asked if you want to preserve the database of Cryptographic Material. If you respond y, the directory containing the database will not be removed during the uninstall.

You will also see this prompt:

```
Are you absolutely sure you want to remove users' OGFS home and audit
directories? (home and audit directories will only be removed if they are
stored on the Software Repository server) (y/n)?
```

Select y if you want to remove the OGFS home and audit directories. If you press n, the directories will not be removed. If you chose to place the OGFS home and audit directories on a server other than the server hosting the Software Repository, the uninstall will not remove those directories even if you press y.

If you installed the core using Custom Mode, it is important that you uninstall the components in the reverse order that they were installed.

8 After the uninstall has completed, remove the `/var/opt/opsware/install_opsware` directory.

If you specified during the uninstall that you want to preserve the database of cryptographic material, you should *not* delete the `/var/opt/opsware/crypto` directory. This directory contains the database of cryptographic material.

Decommission a Facility using the SAS Web Client

Performing this procedure does not shut down or uninstall SA in a facility. Decommission facilities with care, because this task cannot be undone.

When you decommission a facility, the facility is still listed in the SAS Web Client, however, it is grayed out. After a short name is used, even if it is decommissioned, that name cannot be reused.

To decommission a facility, perform the following steps:

1 In the SAS Web Client, deactivate the server running the core for the facility that you want to decommission. (For instructions, see “Deactivating a Server” in the *SA User Guide: Server Automation*.)

2 From the navigation panel, click **Environment** ‰ **Facilities**. The Facilities page appears.

3 Select the facility that you want to decommission.

4 On the Properties tab, note the answer to the following question:

```
Is this facility in use?
```

If the answer is No, the **Decommission** button is displayed.

5 Click **Decommission**.

A Oracle Setup for the Model Repository

This appendix explains how to install, configure, and maintain an Oracle database to support the Model Repository.

Oracle RDBMS Install Basics

The Model Repository is an SA Core Component that stores information in an Oracle database. It stores the following information:

- database users
- database user privileges
- schema information
- baseline data

The SA distribution Media includes a separate Oracle 11g RDBMS software and database installation dual layer DVD. You can simply mount this DVD on the server you plan to use to host the Model Repository and run the installation. See Installing the HP-Supplied Oracle RDBMS Software and Database on page 195 in this chapter for information about the installation steps

You can also use the Oracle Universal Installer to manually install an Oracle 10g or 11g database, however, you will need to perform certain tasks that the HP-supplied database performs automatically on installation. If you plan to use an existing database installation, you must ensure that the database is configured correctly for use with the SA Model Repository.

If you plan to use the Oracle Universal Installer to install the Oracle RDBMS software and database, or will use an existing Oracle database, then you should read the following sections:

- Supported Oracle Versions on page 188
- Hardware Requirements on page 189
- Operating System Requirements on page 191
- The SA Installer HP-Supplied RDBMS Installation Process on page 197
- Pre-Installation Tasks (Oracle Universal Installer) on page 199
- Manually Creating the Oracle Database on page 201
- Post-Oracle Installation Tasks on page 205

The Oracle database must be created before you install the Model Repository.

Supported Oracle Versions

Support for the Model Repository is limited to certain versions of Oracle running on certain versions of operating systems. HP strongly recommends that you also apply the latest Oracle CPU patches. For manual installations, SA supports both the Oracle Standard Edition, Standard Edition One, and the Oracle Enterprise Edition. Table 42 lists the supported Oracle versions.

Table 42 Supported Operating Systems and Oracle Versions

Operating System	Supported Oracle Versions (Standard or Enterprise Edition)
SunOS 10 x86_64	10.2.0.2, 10.2.0.4, 11.1.0.7
Red Hat Enterprise Linux AS 3 x86_32	10.2.0.2, 10.2.0.4
Red Hat Enterprise Linux AS 4 x86_64	10.2.0.2, 10.2.0.4, 11.1.0.7
Red Hat Enterprise Linux AS 5 x86_64	10.2.0.4, 11.1.0.7
SUSE Linux Enterprise Server 10 x86_64	10.2.0.4, 11.1.0.7

Oracle 10.2.0.3 is not supported by SA due to known incompatibilities.

Multiple Oracle Versions and Multimaster Cores

For the database export to succeed during the installation of a Multimaster core, the version of the target database cannot be 10.x if the source database is 11.x. Table 43 lists these allowed version combinations.

Table 43 Database Versions Allowed for Multimaster

Source Database Version	Target Database Version	Allowed?
10	10	Y
10	11	Y
11	10	N
11	11	Y

Hardware Requirements

The server that will host the Oracle database for the Model Repository must meet the hardware requirements listed in this section.

Linux Requirements

The following are hardware requirements for running Oracle 11g under Linux:

- The recommended physical memory is 8 GB. An HP-supplied Oracle installation will use around 1.6-1.7GB of memory. The Oracle SGA memory can be increased after database installation. You can use the following command to check memory status:

  ```
  grep MemTotal /proc/meminfo
  ```

- Recommended swap space: between 1-2 GB RAM, (1.5 times the size of RAM, between 2-8 GB RAM, equal to the size of RAM).

  You can use the following command to check swap space:

  ```
  grep SwapTotal /proc/meminfo
  ```

- Recommended shared memory available required for automatic memory management should be greater than 1GB.

  You can use the following command to check available shared memory:

  ```
  df -k /dev/shm/
  ```

- Free `tmp` space should be 400MB or more

  You can use the following command to check `tmp` space:

  ```
  df -k /tmp | grep / | awk '{ print $3 }'
  ```

- Required Kernel version:

  — Red Hat AS 4: 2.6.9 or later

  — Red Hat AS 5: 2.6.18 or later

  — SUSE Linux 10: 2.6.16.21 or later

  You can use the following command to check the kernel versions:

  ```
  uname -r
  ```

  You can use the following command to check the platform:

  ```
  uname -mi
  ```

  You can use the following command to check the processor type:

  ```
  grep "model name" /proc/cpuinfo
  ```

Solaris Requirements

The following are hardware requirements for running Oracle 11g under Solaris:

- The recommended physical memory is 8 GB. An HP-supplied Oracle installation will use around 1.6-1.7GB of memory. The Oracle SGA memory can be increased after database installation.

  You can use the following command to check the physical memory:

```
/usr/sbin/prtconf | grep "Memory"
```

- Recommended swap space: between 1-2 GB RAM, (1.5 times the size of RAM, between 2-8 GB RAM, equal to the size of RAM).

  You can use the following command to check the swap space:

  ```
  /usr/sbin/swap -s
  ```

- Free `tmp` space should be 400MB or more

  You can use the following command to check `tmp` space:

  ```
  df -k /tmp | grep / | awk '{ print $3 }'
  ```

- Required operating system version is: 5.10

  You can use the following command to check the operating system version:

  ```
  uname -a
  ```

- System architecture should show `64-bit sparcv9 kernel` modules

  You can use the following command to check system architecture:

  ```
  isainfo -kv
  ```

Model Repository (Database) Disk Space Requirements

Additional disk space is required for the Oracle software and the Model Repository data files. Keep in mind that storage requirements for the database grow as the number of managed servers and database activity grows.

As a benchmark figure, you should allow an additional 3.1 GB of database storage for every 1,000 servers in the facility that SA manages. When sizing the tablespaces, follow the general guidelines described in Table 44. If you need to determine a more precise tablespace sizing, contact your technical support representative.

Table 44 Tablespace Sizes

Tablespace	Minimum Size
AAA_DATA	256 MB
AAA_INDX	256 MB
AUDIT_DATA	256 MB
AUDIT_INDX	256 MB
LCREP_DATA	1,500 MB
LCREP_INDX	800 MB
TRUTH_DATA	700 MB
TRUTH_INDX	400 MB
STRG_DATA	700 MB
STRG_INDX	400 MB

Hostname Setup

You must be able to ping the database server hostname. To verify this, enter the following command:

Linux/Red Hat Linux

```
# ping <hostname>
```

or, on the database server, enter the following command:

```
# hostname
```

SUSE Linux

```
# hostname -f
```

If the hostname is not configured correctly, Oracle will not start and you will encounter the following error:

```
ORA-00600: internal error code, arguments: [keltnfy-ldmInit], [46], [1],
[], [], [], [], []
```

Operating System Requirements

The following sections list the operating system requirments for Oracle 11g. The SA Installer performs an automated check to ensure that these requirements are met on the Oracle host before proceeding with the installation of the Oracle 11g software and database.

If you create the database using the Oracle Universal Installer rather than the SA Installer, you must check for these packages and patches manually.

Required Packages for Red Hat Enterprise Linux AS 4 x_64

The following packages are required for Oracle 11g on Red Hat Enterprise Linux AS 4 64 x86_64. These packages must be the versions listed or higher.

Table 45 Packages Required by Oracle 11g under Red Hat Enterprise Linux AS 4 x86_64

Required Packages	Architecture	Version
binutils	x86_64	2.15.92.0.2
compat-libstdc++-33	i386	3.2.3
compat-libstdc++-33	x86_64	3.2.3
elfutils-libelf	x86_64	0.97
elfutils-libelf-devel	x86_64	0.97
gcc	x86_64	3.4.5
gcc-c++	x86_64	3.4.5
glibc	i686	2.3.4-2.19
glibc	x86_64	2.3.4-2.19

Table 45 Packages Required by Oracle 11g under Red Hat Enterprise Linux AS 4 x86_64

Required Packages	Architecture	Version
glibc-common	x86_64	2.3.4
glibc-devel	i386	2.3.4
glibc-devel	x86_64	2.3.4
libaio	i386	0.3.105
libaio	x86_64	0.3.105
libaio-devel	x86_64	0.3.105
libgcc	i386	3.4.5
libgcc	x86_64	3.4.5
libstdc++	x86_64	3.4.5
libstdc++	i386	3.4.5
libstdc++-devel	x86_64	3.4.5
make	x86_64	3.80
sysstat	x86_64	5.0.5

Required Packages for Red Hat 5 Server x86_64

The following packages are required for Oracle 11g on Red Hat Enterprise Linux 5 Server x86_64. These packages must be the versions listed or higher.

Table 46 Required Packages for Red Hat 5 Server x86_64

Required Packages	Architecture	Version
binutils	x86_64	2.17.50.0.6
compat-libstdc++-33	i386	3.2.3
compat-libstdc++-33	x86_64	3.2.3
elfutils-libelf	x86_64	0.125
elfutils-libelf-devel	x86_64	0.125
gcc	x86_64	4.1.1
gcc-c++	x86_64	4.1.1
glibc	i686	2.5-12
glibc	x86_64	2.5-12
glibc-common	x86_64	2.5
glibc-devel	i386	2.5-12
glibc-devel	x86_64	2.5

Table 46 Required Packages for Red Hat 5 Server x86_64

Required Packages	Architecture	Version
libaio	i386	0.3.106
libaio	x86_64	0.3.106
libaio-devel	x86_64	0.3.106
libgcc	i386	4.1.1
libgcc	x86_64	4.1.1
libstdc++	i386	4.1.1
libstdc++	x86_64	4.1.1
libstdc++-devel	x86_64	4.1.1
make	x86_64	3.81
sysstat	x86_64	7.0.0

Required Packages for SUSE Linux Enterprise Server 10 x86_64

The following packages are required for Oracle 11g on SUSE Linux Enterprise Server 10 x86_64. These packages must be the versions listed or higher

Table 47 Packages Required by Oracle 11g under Suse Linux Enterprise Server 10 x86_64

Required Packages	Architecture	Version
binutils	x86_64	2.16.91.0.5
compat-libstdc++	x86_64	5.0.7-22.2
gcc	x86_64	4.1.0
gcc-c++	x86_64	4.1.0
glibc	x86_64	2.4-31.2
glibc-32bit	x86_64	2.4-31.2
glibc-devel	x86_64	2.4
glibc-devel-32bit	x86_64	2.4
libaio	x86_64	0.3.104
libaio-32bit	x86_64	0.3.104
libaio-devel	x86_64	0.3.104
libelf	x86_64	0.8.5
libgcc	x86_64	4.1.0
libstdc++	x86_64	4.1.0

Table 47 Packages Required by Oracle 11g under Suse Linux Enterprise Server 10 x86_64

Required Packages	Architecture	Version
libstdc++-devel	x86_64	4.1.0
make	x86_64	3.80
sysstat	x86_64	6.0.2

To verify whether these RPMs are installed on the OS, enter the following command:

```
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' <rpm_name>
```

Required Packages for Solaris 10

The following packages are required for Oracle 11g on Solaris 10 servers. These packages must be the versions listed or higher:

Oracle 11g

Solaris 10 must have the following packages for Oracle 11g:

```
SUNWarc
SUNWbash
SUNWbtool
SUNWhea
SUNWlibC
SUNWlibm
SUNWlibms
SUNWsprot
SUNWtoo
SUNWi1of
SUNWi1cs
SUNWi15cs
SUNWxwfnt
SUNWpool
SUNWpoolr
```

Required Patches for Solaris 10

The following patches are required on Solaris 10 for Oracle 11g:

Table 48

Patch	Version	Comment
127111	02	`#libc patch`
137111	04	`#kernel patch`

Required Patch for Manual Oracle 11g Installations

If you plan to install Oracle 11g (for the Model Repository) using the Oracle Universal Installer or use an existing Oracle 11g installation, you must ensure that Oracle Bug Patch 8300752 has been applied to that database. This patch has already been applied to the HP-supplied Oracle RDBMS.

Oracle 10g on Solaris 10 Servers

When Oracle 10.2 is installed on T2000 hardware with the Solaris 10 operating system, the SA Installer hangs during the installation of the Model Repository. The Oracle alert.log includes errors, such as the following:

```
MMNL absent for 28552 secs; Foregrounds taking over
Wed Aug 2 12:45:57 2006
MMNL absent for 28853 secs; Foregrounds taking over
Wed Aug 2 12:50:57 2006
MMNL absent for 29151 secs; Foregrounds taking over
```

Customers should look at Bug 6385446 from Sun Microsystems and apply Patches 118833-18, 119578-24 and 119254-24 as per:

http://sunsolve.sun.com/search/document.do?assetkey=1-26-102289-1

Installing the HP-Supplied Oracle RDBMS Software and Database

If you are manually installing the Oracle RDBMS software and database, skip this section and go to Pre-Installation Tasks (Oracle Universal Installer) on page 199.

The SA distribution Media includes an Oracle 11g database installation DVD. You can simply mount this DVD on the server you plan to use to host the Model Repository and enter the following command to begin the Oracle installation.

```
/<distro>/oracle_sas/install_opsware.sh --verbose
```

You will see a screen similar to this:

```
Install Type: "Oracle RDBMS for SAS"

Please select the interview mode. Simple mode uses default values for many of
the configuration parameters. Advanced mode allows you to fully configure the
installation.

1 - Simple Interview Mode

2 - Advanced Interview Mode


Please select the interview mode from the menu, type 'h' for help, 'q' to
quit: 1

The Opsware Installer will now interview you to obtain the installation
parameters it needs. You can use the following keys to navigate forward
and backward through the list of parameters:
```

```
Control-P - go to the previous parameter
Control-N - go to the next parameter
Return - accept the default (if any) and go to the next parameter
Control-F - finish parameter entry
Control-I - show this menu, plus information about the current parameter

Press Control-F when you are finished. The Opsware Installer will perform
a final validation check and write out a response file that will be
used to install the Opsware components.
```

Enter `1` for the Simple Interview or `2` for the Advanced Interview (which allows you to modify additional parameters, if necessary). For example, if you choose the Simple Interview, you will see a screen similar to this:

```
Parameter 1 of 3 (truth.oaPwd)Please enter the password for the opsware_admin
user. This is the password used to connect to the Oracle database.:
opsware_admin
Validating... OK.

Parameter 2 of 3 (truth.servicename)Please enter the service name (aka TNS
name) of the Model Repository instance in the facility where Opsware Installer
is being run [truth]:
Validating... OK.

Parameter 3 of 3 (truth.port)Please enter the port on which the Model
Repository database is listening. [1521]:
Validating... OK.

All parameters have values.  Do you wish to finish the interview? (y/n): y


Concluding interview.

Interview complete.
```

You are then asked to supply the name of the response file in which to store your interview responses. The default is `oiresponse.oracle.sas`:

```
Name of response file to write [/usr/tmp/oiresponse.oracle_sas]:
Response file written to /usr/tmp/oiresponse.oracle_sas.

Would you like to continue the installation using this response file? (y/n): y
```

You will then see this screen:

```
Welcome to the Opsware Installer.
Please select the components to install.
1 ( ) Oracle RDBMS for SAS
Enter a component number to toggle ('a' for all, 'n' for none).
When ready, press 'c' to continue, or 'q' to quit.

Selection:
```

Select `1` and press `c` to begin the database installation. When the installation is complete, a message to that effect is displayed. You can now continue with the SA installation.

The SA Installer HP-Supplied RDBMS Installation Process

The SA Installer uses the following process when installing the HP-supplied Oracle RDBMS .

SA Installer Changes to Database Configuration and Files

When you install the HP-supplied Oracle RDBMS from the SA Installer Oracle installation option, the installer

- Checks that all requirements are met on the host server (see Hardware Requirements on page 189 and Operating System Requirements on page 191).
- Sets certain kernel parameters to required values. For details about these parameter changes, see Kernel Parameter Values on page 198.
- Creates the Unix user `oracle` locally in `/etc/passwd`.
- Creates the Unix groups `dba` and `oinstall` locally in `/etc/group`.
- Sets the `$ORACLE_HOME` environment variable to the following directory:

  `/u01/app/oracle/product/11.1.0/db_1`
- Sets the `$ORACLE_SID` environment variable to `truth`.
- Creates an Oracle instance with the required `init.ora` parameters.
- Creates the tablespaces and data and index files under the following directories:

  ```
  /u01/oradata/truth
  /u02/oradata/truth
  /u03/oradata/truth
  /u04/oradata/truth
  ```
- The system administrator can configure the `/u01`, `/u02`, `/u03`, `/u04` directories before installing the Oracle RDBMS software.
- Gets the service name (TNS name) from the SA Installer interview (`truth.servicename` prompt) and inserts it into the `tnsnames.ora` file in `$ORACLE_HOME/network/admin` and `/var/opt/oracle`. The SA Installer changes the value of the `host` parameter to the value returned by the Unix `hostname` command.
- In the `/$ORACLE_HOME/network/admin/listener.ora` file, changes the value of the `host` parameter to the value returned by the Unix `hostname` command.

  The listener is password protected and OS authenticated. (The default password is `opsware`.) By default, it listens on port 1521.
- Creates the `/etc/init.d/opsware-oracle` script, which you can use to start up and shut down the database and listener.

  This script is linked to corresponding scripts in the `/etc/rc*.d` directories..
- Creates the user `opsware_admin` with the required privileges.
- After installation is complete, you can examine the logs that are created here:

  `/var/log/opsware/install_opsware`

Kernel Parameter Values

During the SA Installer installation of the HP-supplied Oracle RDBMS , the installation also sets the values for certain parameters in various files. If you manually install the Oracle database, or use an existing database, you must insure that these values are specified correctly. This section lists the parameters set by the installer that can be changed without adversely affecting SA.

Changing Kernel Parameter Values for Linux

This section identifies the kernel parameters you can change for Red Hat Linux Enterprise Server AS 4 x86_64, Red Hat Linux Enterprise Server AS 5 x86_64, and SUSE Linux Enterprise Server 10 x86_64.

You can change values for the following parameters in `/etc/sysctl.conf`:

```
kernel.shmmax=2147483648
kernel.shmall=2097152
kernel.shmmni=4096
kernel.sem=250 32000 100 128
net.core.rmem_default=4194304
net.core.wmem_default=262144
net.core.rmem_max=4194304
net.core.wmem_max=262144
fs.file-max=65536
net.ipv4.ip_local_port_range=1024 65000
net.ipv4.tcp_wmem=262144 262144 262144
net.ipv4.tcp_rmem=4194304 4194304 4194304
```

You can change values for the following parameters in `/etc/security/limits.conf`:

```
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft nproc 2047
oracle hard nproc 16384
```

You can change values for the following parameters in `/etc/pam.d/login`:

```
session required /lib/security/pam_limits.so
```

You can change values for the following parameters in `/etc/fstab`:

```
shmfs /dev/shm tmpfs size=4g 0
```

Additional Modifiable SUSE Kernel Parameter Values

This section identifies additional required settings for SUSE Linux Enterprise Server 10 x86_64 when running Oracle 11g:

- Enter the following command to cause the system to read the `/etc/sysctl.conf` file when it restarts:

  ```
  # /sbin/chkconfig boot.sysctl on
  ```

- You must enter the GID of the `oinstall` group as the value for the parameter `/proc/sys/vm/hugetlb_shm_group`. Doing this grants members of `oinstall` a group permission to create shared memory segments.

For example, where the oinstall group GID is 501:

```
# echo 501 > /proc/sys/vm/hugetlb_shm_group
```

After running this command, use vi to add the following text to /etc/sysctl.conf, and enable the boot.sysctl script to run on system restart:

```
vm.hugetlb_shm_group=501
```

Only one group can be defined as the `vm.hugetlb_shm_group`.

Changing Kernel Parameter Values for Solaris 10

To change a kernel parameter for Solaris 10, perform the following steps:

1. Enter set `noexec_user_stack=1` in `/etc/system`.
2. Run the following commands:

```
projadd -U oracle -K "project.max-shm- memory=(priv,2048MB,deny)"
user.oracle

projmod -s -K "project.max-sem-ids=(priv,100,deny)" user.oracle
projmod -s -K "process.max-sem-nsems=(priv,256,deny)" user.oracle

projmod -s -K "project.max-shm-ids=(priv,100,deny)" user.oracle

echo "oracle::::project=user.oracle"  >> /etc/user_attr
```

3. Use the `vi` editor for `/etc/project` and `/etc/user_attr` to verify the changes made in step 2.

Pre-Installation Tasks (Oracle Universal Installer)

If you plan to install the HP-supplied Oracle RDBMS software and database using the SA Installer, you do not need to perform the tasks in this section. The tasks in this section are only for Oracle databases installed using the Oracle Universal Installer and are required for compatibility with SA.

This section discusses the prerequisites for an installation of the Oracle RDBMS using the Oracle Universal Installer for use with SA. For more detailed information about installing Oracle, see the *Oracle Installation Guide* for your operating system. Each operating system and Oracle version has a different guide. The Oracle documentation is available at the following URL:

```
http://www.oracle.com/technology/documentation/index.html
```

Before installing the Oracle RDBMS software, perform the following steps:

The sample files referenced in these steps can be obtained from your HP Support representative. See Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files on page 201.

1. Verify that the server has the hardware and software listed in Hardware Requirements on page 189 and Operating System Requirements on page 191.
2. Obtain the sample files and unzip them.

3 Set the kernel parameters.

The easiest way to set these parameters is by copying and editing the following sample files:

```
kernel_params_redhat.txt
kernel_params_solaris.txt
```

These two files contain instructions, Unix commands, and lines of text for configuration files.

4 Create the required Unix users and groups by running the following commands. (If you use a directory different than `/u01/app/oracle`, modify the commands accordingly):

```
mkdir -p /u01/app/oracle
groupadd oinstall
groupadd dba
groupadd dboper
useradd -g oinstall -G dba \
  -d /u01/app/oracle -s /usr/bin/sh oracle
chown oracle:oinstall /u01/app/oracle
Set the environment variables for the oracle user.
```

The easiest way to set these variables is by obtaining and editing the following sample files:

```
bash_profile
profile
```

Now you should be ready to install the Oracle RDBMS. For instructions, see the *Oracle Installation Guide* for your operating system.

Baseline Data Installation

The following steps are required for Red Hat Enterprise Linux AS 4 x86_64, Red Hat Enterprise Linux AS 5 x86_64, Sun Solaris x86_64, and SUSE Enterprise Linux 10 x86_64 because, during a Model Repository fresh install, baseline data is not inserted completely. Oracle does not insert some of the baseline data in `role_classes` and other tables and there are no errors, failures or trace files generated by Oracle. This is a silent failure and an intermittent problem. The Model Repository installs successfully because there are no error messages returned from Oracle, but later the Data Access Engine (Spin) install fails due to missing baseline data.

Before installing Oracle on Red Hat Enterprise Linux AS 4 x86_64, Red Hat Enterprise Linux AS 5 x86_64, Sun Solaris x86_64, or SUSE Enterprise Linux 10 x86_64 on the Model Repository (`truth`)/database host, run the following commands:

```
# Su - oracle
# Sqlplus "/ as sysdba"

SQL> ALTER SYSTEM SET EVENT='12099 trace name context forever,
level 1' SCOPE=SPFILE;
SQL> Shutdown immediate;
SQL> Startup
SQL> Exit
```

Now, you can run the SA Installer and install the Model Repository.

Oracle XDB Component Installation Requirements

During a Multimaster installation, SA exports the database using Oracle's Export utility. Due to an Oracle bug, the Export utility fails if the XDB component is installed *and* NLS_LENGTH_SEMANTICS=CHAR (as required for SA). To avoid this error, you must install Oracle excluding the XDB component.

Manually Creating the Oracle Database

If you will install the HP-supplied Oracle RDBMS software and database, you do not need to perform the tasks in this section. The tasks in this section are only for Oracle installed using the Oracle Universal installer and are required for compatibility with SA.

When the SA Installer installs the Oracle RDBMS software and database, it runs certain scripts that do configuration tasks, create users, set password and parameter values, etc.

When you manually install the Oracle RDBMS, certain of these scripts must be run, others are optional (you can manually make the required modifications to Oracle settings for SA or you can run the script `truth.sh` which will automatically run all the required scripts in the correct order).

The SQL scripts that must be run or edited are:

- `CreateDB.sql`
- `CreateDBFiles.sql`
- `CreateUserOpsware_Admin.sql`
- `init.ora`

Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files

The following describes the script files, SQL scrips, and configuration files that are run or edited when you run the truth.sh script. These files are available from your HP Support representative.

- **truth.sh**: A shell script that creates directories and then launches the `truth.sql` script. Running this script causes all the tasks performed in this list to be performed automatically.
- **truth.sql**: Prompts for passwords of the SYS and SYSTEM users and then launches the remainder of the SQL scripts in this list.
- **CreateDB.sql**: Creates a database with the UTF8 character set (as required by SA), the data and index files, the default temporary tablespace, the undo tablespace, and the log files.

The database must have the character set UTF8 available.

- **CreateDBFiles.sql**: Creates the following tablespaces that are required by SA:

  ```
  LCREP_DATA
  LCREP_INDX
  TRUTH_DATA
  TRUTH_INDX
  AAA_DATA
  AAA_INDX
  AUDIT_DATA
  AUDIT_INDX
  STRG_DATA
  STRG_INDX
  ```

  See Tablespace Sizes on page 40 in Chapter 2 for information about for additional tablespace sizing information.

- **CreateDBCatalog.sql**: Runs Oracle scripts to create data system catalog objects.

- **JServer.sql**: Sets up the Oracle Java environment.
- **CreateAdditionalDBFiles.sql**: Adds data and index files to certain tablespaces and allocates additional disk space. This script is optional, but recommended.
- **CreateUserOpsware_Admin.sql**: Creates the `opsware_admin` database user and grants permissions (privileges) to this user (required by SA).
- **postDBCreation.sql:** Creates the `spfile` from the `pfile` (parameter file).
- **init.ora**: Contains initialization parameters for the database. Certain parameter values are required by SA. See Required and Suggested Parameters for init.ora on page 203.
- **tnsnames.ora**: Enables resolution of database names used internally by SA.
- **listener.ora**: Contains configuration parameters for the listener. SA by default listens on port 1521. You can change the default port during installation or by editing the tsnames.ora file.
- **bash_profile**: Sets environment variables and sets shell limits for the `oracle` Unix user.
- **profile**: Sets environment variables for the `oracle` Unix user.
- **kernel_params_redhat.txt**: Contains kernel parameters for Red Hat Enterprise Linux 3 AS.
- **kernel_params_solaris.txt**: Contains kernel parameters for Solaris 10.
- **opsware-oracle**: A script residing in `/etc/init.d` that starts up and shuts down the database and listener.

  Note that the `/etc/init.d/opsware-sas` script, which starts and stops the SA components, does not start and stop the database and listener. For more information on the `opsware-sas` script, see the *SA Administration Guide*.
- **Export-Import**: A directory that contains parameter files and instructions for performing full database exports and imports.

Files that Must be Run or Edited for a Manual Oracle Installation

Even if you plan to configure your Oracle installation manually and not run `truth.sh` to automatically configure the Oracle installation, the following scripts must be run and `init.ora` must have certain parameter values edited or added as shown in Required and Suggested Parameters for init.ora on page 203.

- **CreateDB.sql**: Creates a database with the UTF8 character set (as required by SA), the data and index files, the default temporary tablespace, the undo tablespace, and the log files.
- **CreateDBFiles.sql**: Creates the following tablespaces that are required by SA:

  ```
  LCREP_DATA
  LCREP_INDX
  TRUTH_DATA
  TRUTH_INDX
  AAA_DATA
  AAA_INDX
  AUDIT_DATA
  AUDIT_INDX
  STRG_DATA
  STRG_INDX
  ```

  See Tablespace Sizes on page 40 in Chapter 2 for information about for additional tablespace sizing information.
- **CreateUserOpsware_Admin.sql**: Creates the `opsware_admin` database user and grants permissions (privileges) to this user (required by SA). If you plan to create the opsware_admin without running this script, see Create the User Opsware_Admin on page 203.

- **init.ora**: Must be edited as shown in Required and Suggested Parameters for init.ora on page 203.

Create the User Opsware_Admin

The following explains how to create this user:

To create the `opsware_admin` user after a manual Oracle installation, log in to SQL*Plus and enter the following:

```
# Su - oracle
# Sqlplus "/ as sysdba"

SQL> create user opsware_admin identified by opsware_admin
        default tablespace truth_data
        temporary tablespace temp
        quota unlimited on truth_data;

SQL> grant alter session to opsware_admin with admin option;
grant create procedure to opsware_admin with admin option;
grant create public synonym to opsware_admin with admin option;
grant create sequence to opsware_admin with admin option;
grant create session to opsware_admin with admin option;
grant create table to opsware_admin with admin option;
grant create trigger to opsware_admin with admin option;
grant create type to opsware_admin with admin option;
grant create view to opsware_admin with admin option;
grant delete any table to opsware_admin with admin option;
grant drop public synonym to opsware_admin with admin option;
grant select any table to opsware_admin with admin option;
grant select_catalog_role to opsware_admin with admin option;
grant query rewrite to opsware_admin with admin option;
grant restricted session to opsware_admin with admin option;

grant execute on dbms_utility to opsware_admin with grant option;
grant analyze any to opsware_admin;
grant insert, update, delete, select on sys.aux_stats$ to opsware_admin;
grant gather_system_statistics to opsware_admin;
grant create job to opsware_admin;

grant alter system to opsware_admin;
grant create role to opsware_admin;
grant create user to opsware_admin;
grant alter user to opsware_admin;
grant drop user to opsware_admin;
grant create profile to opsware_admin;
grant alter profile to opsware_admin;

grant drop profile to opsware_admin;
```

Required and Suggested Parameters for init.ora

(*Both Oracle 10g and 11g)* For SA, the following `init.ora` entries are either suggested or required:

```
log_buffer>=1048576
db_block_size>=8192
open_cursors >=1000
```

```
session_cached_cursors=50
nls_length_semantics=CHAR
nls_sort=GENERIC_M
processes >=1024
undo_management=AUTO (Suggested)
undo_tablespace=UNDO (Suggested)
query_rewrite_integrity=TRUSTED
query_rewrite_enabled=true
optimizer_mode=choose or all_rows
optimizer_index_cost_adj=20
optimizer_index_caching=80
cursor_sharing=SIMILAR (value can be set to SIMILAR(preferred) or EXACT,
recommended only if you encounter an Oracle error)
recyclebin=OFF
event="12099 trace name context forever, level 1"
_complex_view_merging=false
```

(*Oracle 10g only*) For SA, the following `init.ora` entries are either suggested or required:

```
sga_max_size >=1GB
db_cache_size>=629145600
shared_pool_size>=262144000
java_pool_size>=52428800
large_pool_size>=52428800
job_queue_processes >=10
sessions >=1152
pga_aggregate_target >=104857600
workarea_size_policy=auto
remote_login_passwordfile=SHARED
```

(*Oracle 11g only*) For SA, the following `init.ora` entries are either suggested or required:

```
memory_target >= 1.7GB
job_queue_processes >=1000 (default)
remote_login_passwordfile=EXCLUSIVE
```

Creating the Database using the HP-Supplied Scripts

To create the Oracle database using the HP-supplied scripts, perform the following steps:

1. Obtain the database creation scripts from your HP support representative..
2. Configure the scripts. See Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files on page 201
3. Log in to the server as the Unix user `oracle`.
4. Copy the HP-supplied `init.ora` file to the following directory:

   `$ORACLE_BASE/admin/truth/create`
5. Examine the SQL scripts that you will run in **step 7**. If necessary, edit the scripts to conform to your organization's policies.
6. Log on to the server as the `oracle` user and change the mode of the HP-supplied `truth.sh` script:

   `chmod 755 truth.sh`
7. Launch the SQL scripts that create the database by running the `truth.sh` script:

   `./truth.sh`

8 After the scripts launched by `truth.sh` complete, check the log files in the following directory for errors:

`/u01/app/oracle/admin/truth/scripts/*.log`

Post-Oracle Installation Tasks

If you will install the HP-supplied Oracle database, you do not need to perform the tasks in this section. The tasks in this section are only for Oracle databases installed using the Oracle Universal installer and are required for compatibility with SA.

After creating the database, but before installing the Model Repository with the SA Installer, perform the following steps:

1 Create the `tnsnames.ora` file in the following directory:

`$ORACLE_HOME/network/admin`

Verify that the file conforms to the rules listed in tnsnames.ora File Requirements on page 206.

2 If it does not exist, create the following directory:

```
mkdir -p /var/opt/oracle
```

3 Create the following symbolic link:

```
ln -s $ORACLE_HOME/network/admin/tnsnames.ora \
/var/opt/oracle/tnsnames.ora
```

4 Make sure that the oracle Unix user has read-write permission on the `tnsnames.ora` file.

5 For Red Hat Enterprise Linux 3 AS, create another symbolic link:

```
ln -s /etc/oratab /var/opt/oracle/oratab
```

6 Copy the sample `opsware-oracle` script to `/etc/init.d/`.

7 Link `/etc/init.d/opsware-oracle` to corresponding scripts in the `/etc/rc*` directories. For example:

```
ln -s /etc/init.d/opsware-oracle \
      /etc/rc0.d/K02opsware-oracle
ln -s /etc/init.d/opsware-oracle \
      /etc/rc1.d/K02opsware-oracle
ln -s /etc/init.d/opsware-oracle \
      /etc/rc2.d/S60opsware-oracle
ln -s /etc/init.d/opsware-oracle \
      /etc/rcS.d/K02opsware-oracle
```

8 Copy the sample `listener.ora` file to `$ORACLE_HOME/network/admin`.

9 In `listener.ora`, change the value of the `host` parameter to the host name of server running the database.

Location of Additional Oracle Data Files

If you want to add data files to a database created with the SA Installer, you can add them to the following directories:

`/u01/oradata/truth`

```
/u02/oradata/truth
/u03/oradata/truth
/u04/oradata/truth
```

tnsnames.ora File Requirements

The `tnsnames.ora` file enables resolution of database names used internally by the core components. SA has the following requirements for the `tnsnames.ora` file:

- The file must reside in the following location:

  `/var/opt/oracle/tnsnames.ora`

- If the core is installed across multiple servers, a copy of the file must reside on the servers hosting the following components:
  - Model Repository
  - Infrastructure Component bundle (required by the Data Access Engine, Model Repository Multimaster Component, Software Repository Store)
  - Slice Component bundle (required by the Command Center, Web Services Data Access Engine, Global File System)
- For a core installed on multiple servers, the directory path of the `tnsnames.ora` file must be the same on each server.
- In a Single Core installation, the `tnsnames.ora` file must contain an entry for the Model Repository, as in the following example:

```
truth =
(DESCRIPTION=
(ADDRESS=(HOST=magenta.example.com)(PORT=1521)
(PROTOCOL=tcp))
(CONNECT_DATA=(SERVICE_NAME=truth)))
```

tnsnames.ora: Multimaster Mesh Requirements

In a Multimaster Mesh, the `tnsnames.ora` file must be set up for a Source Core and a Destination Core using the following guidelines.

Source Core

The `tnsnames.ora` file must contain an entry for its own Model Repository. The port number must be set to the port that you have designated that the Oracle listener process use, such as `1521 (default)`, `1526`, and so on.

The `tnsnames.ora` file must also contain an entry that specifies the Source Core Management Gateway. This port is used by the Data Access Engine for Multimaster traffic. The port number is derived from the following formula: (20000) + (facility ID of the Destination Core).

Example: In the following example, the TNS service name of the Source Core is `orange_truth`, which runs on the host `orange.example.com`. The TNS name of the Destination Core is `cyan_truth`, which has a facility ID of `556`. Note that the entry for `cyan_truth` specifies `orange.example.com`, which is the host running the Source Core's Management Gateway.

```
orange_truth=(DESCRIPTION=(ADDRESS=(HOST=orange.example.com)(PORT=1521)
(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
cyan_truth=(DESCRIPTION=(ADDRESS=(HOST=orange.example.com)(PORT=20556)
(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

Destination Core

The `tnsnames.ora` file must contain an entry for its own Model Repository. The port number must be set to the port that you have designated that the Oracle listener process use, such as `1521` (default), `1526`, and so on. The `tnsnames.ora` file does not require any entries for other cores in the mesh.

Example: In the following example, the TNS service name of the Destination Core is `cyan_truth`, and the core runs on the host, `cyan.example.com`.

```
cyan_truth=(DESCRIPTION=(ADDRESS=(HOST=cyan.example.com)(PORT=1521)
(PROTOCOL=tcp))(CONNECT_DATA=(SERVICE_NAME=truth)))
```

Requirements for Enabling Oracle Daylight Saving Time (DST)

To enable Daylight Saving Time for the Oracle database, you must apply database tier patches. To apply these patches, perform the following steps:

1. Verify that your database is running on Oracle 10g or higher. If you are on an earlier database release, use one of the following MetaLink Notes to upgrade your database:

   10gR2 Database: MetaLink Note 362203.1

2. Use MetaLink Note 359145.1 to apply Oracle Database time zone fixes specific to your database version.

3. Use MetaLink Note 359145.1 to apply time zone fixes to the Oracle Java Virtual Machine (JVM) in the Oracle Database specific to your E-Business Suite database version.

Installing the Model Repository Database on a Remote Server

To install or upgrade the Model Repository Oracle database on a remote server, perform the following steps:

1. Perform the following tasks on the server on which you will run the SA Installer:

   a. Ensure that the hostname `truth` resolves to the remote database server, not to the server on which you'll be running the SA Installer.

   b. Install the Oracle client. For a Multimaster install, you need the full Oracle client; for a Single Core (standalone) installation, the Oracle Instant client will suffice. The client software must be owned by the OS user `oracle`. Install the Oracle client in a location like

      `/opt/opsware/oracle_client_model_repo`.

      — You can copy the Oracle instant client from an existing core. The Oracle Instant client is by default installed under `/opt/opsware/oracle_client` on the Web Services Data Access Engine (`twist`) host.

      — Otherwise, download the appropriate Oracle full client from

        http://www.oracle.com/technology/software/products/database/index.html

   c. (For an Oracle Full client install, this step is not necessary.) Copy the database server's `$ORACLE_HOME/jdbc/lib/classes12.zip` to the client's Oracle home, for example:

      ```
      # scp oracle@truth:$ORACLE_HOME/jdbc/lib/classes12.zip \
      /opt/opsware/oracle_client_model_repo/jdbc/lib
      ```

   d. Copy the database server's `/var/opt/oracle/tnsnames.ora` file to the client host and ensure that the hostname in the file resolves properly.

e Ensure that the SA Installer response file has the correct path to the client `tnsnames.ora` file (`%truth.tnsdir`), oracle client home (`%truth.orahome`), listener port (`%truth.port`), SA Installer machines subdomain (`%truth.dcSubDom`), etc. Based on the above steps your parameter values will be:

- `%truth.tnsdir=/var/opt/oracle`
- `%truth.orahome=/opt/opsware/oracle_client_model_repo`
- `%truth.port=1521`
- `%truth.dcSubDom=prod.example.com`

2 Set up the following on the Model Repository host (Oracle database server):

a Log in as user `oracle`

b cd `$ORACLE_HOME/network/admin`

c Ensure that the `listener.ora` file has the following `SID_LIST_*` section:

```
SID_LIST_<your_listener_name> =
   (SID_LIST =
     (SID_DESC=
       (SID_NAME=truth)
       (ORACLE_HOME=<oracle_home>
     )
   )
```

d Ensure that the listener is started with the command

```
lsnrctl start <your_listener_name>
```

Troubleshooting Remote Model Repository Installation

When you install or upgrade the Model Repository on a remote database server, Oracle gives the following error and the installer aborts:

```
Error: ORA-12526: TNS:listener: all appropriate instances are in restricted
mode
```

Problem

When SA installs or upgrades the schema in the Oracle database, it puts the database in a *restricted mode*. In Oracle 10g and 11g, the standard listener will reject connections if the database is in a restricted mode. In Oracle 10g and 11g, a database administrator can only access the restricted instance locally from the machine that the instance is running on.

Solution

In Oracle10g and 11g, if the listener has the `SID_LIST_*` paragraph in the `listener.ora` file, then the users with *restricted session* privilege are able to connect to a remote database, even if the database is in restricted more. If the `listener.ora` file does not have the `SID_LIST_*` paragraph, then the listener rejects the client connections and gives an `ORA-12526: TNS: listener: all appropriate instances are in restricted mode` error.

Example: A listener.ora Entry

```
OPSCORE1 =
       (DESCRIPTION_LIST =
         (DESCRIPTION =
           (ADDRESS = (PROTOCOL = TCP)(HOST = opscore1.mycompany.com)(PORT = 1521))
           (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
         )
       )

    SID_LIST_OPSCORE1 =
       (SID_LIST =
        (SID_DESC=
            (SID_NAME=truth)
            (ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1)
        )
         (SID_DESC =
           (SID_NAME = PLSExtProc)
           (ORACLE_HOME = /u01/app/oracle/product/10.2.0/db_1)
           (PROGRAM = extproc)
         )
)
```

In this example, the listener alias is `OPSCORE1`.

To start, stop, or check the status of the listener, enter the following commands:

```
# su - oracle
```

To start the listener:

```
> lsnrctl start opscore1
```

To stop the listener:

```
> lsnrctl stop opscore1
```

To check the status of the listener:

```
> lsnrct status opscore1
```

Garbage Collection

The Garbage Collector (GC) is a stored procedure written in PL/SQL that runs in the database on a schedule. The GC procedures look at the `AUDIT_PARAMS` table to determine the retention period to use to delete the old data. The GC PL/SQL procedures are managed by Oracle's `dba_jobs`.

Data Retention Period

When GC runs, it looks at the values in the `AUDIT_PARAMS` table to determine what retention period to use when deleting objects.

The `AUDIT_PARAMS` table is not replicated, so there is a possibility that these retention periods may become out of synch, which can cause severe Multimaster conflict issues. You must ensure that the values in the `AUDIT_PARAMS` table are exactly the same for all the cores in a mesh.

```
# Sqlplus "/ as sysdba"
SQL> col name format a20;
SQL> col value format a20;
SQL> col AUDIT_PARAM_ID format a15;
SQL> select AUDIT_PARAM_ID, NAME, VALUE from audit_params;
```

The parameters from AUDIT_PARAMS table and their default values are:

```
AUDIT_PARAM_ID  NAME                 VALUE
--------------- -------------------- --------------------
2               DAYS_WAY             30         (These are the completed way sessions)
3               DAYS_CHANGE_LOG      180        (These are the server history events)
4               LAST_DATE_WAY        04-APR-09
5               LAST_DATE_CHANGE_LOG 05-NOV-08
6               DAYS_AUDIT_LOG       180        (These are the audit logs)
7               LAST_DATE_AUDIT_LOG  05-NOV-08
```

As of SA 7.80, the DAY_TRAN parameter that controlled retention time for transactions has been removed. To control transaction retention time, instead use the system configuration parameter `vault.garbageCollector.daysToPreserve`.

From the SAS Web Client **Navagation** panel, select **System Configuration** ‰ **Model Repository Multimaster Component** to change the value for the parameter (the default is 7).

Modifying the Retention Period Values

To update the data, run a SQL command similar to the following example as user LCREP:

```
# Su - oracle
# Sqlplus "/ as sysdba"
SQL> grant create session to lcrep;
SQL> connect lcrep/<password>
SQL> update AUDIT_PARAMS set value=30 where name = 'DAYS_AUDIT_LOG';
```

The values in the AUDIT_PARAMS table must be exactly the same for all the cores in a mesh.

Viewing GC DBA_JOBS

When the Model Repository is installed, the SA Installer sets up these jobs, which perform garbage collection.

GC jobs can be viewed by logging in to SQL*Plus and running the following SQL commands:

```
# Su - oracle
# Sqlplus "/ as sysdba"
SQL> col schema_user format a10
SQL> col what format a50
SQL> set line 200
SQL> select job, schema_user, last_date, this_date, next_date, broken,
what from dba_jobs where priv_user= 'GCADMIN';

JOB SCHEMA_USE LAST_DATE THIS_DATE NEXT_DATE B WHAT
--- ---------- --------- --------- --------- - ----------------------------------------
  4 GCADMIN    04-MAY-09           05-MAY-09 N WAYPURGE.GC_SESSIONS;
  5 GCADMIN    04-MAY-09           05-MAY-09 N CHANGELOGPURGE.GC_CHANGELOGS;
```

```
 6 GCADMIN    04-MAY-09          05-MAY-09 N AUDITPURGE.GC_AUDITLOGS;
 7 GCADMIN    04-MAY-09          05-MAY-09 N STORAGEINITIATORPURGE.GC_STORAGEINITIATORS;
```

where:

`WAYPURGE.GC_SESSIONS` - Performs a `sessions` garbage collection

`CHANGELOGPURGE.GC_CHANGELOGS` - Performs a `changelogs` garbage collection

`AUDITPURGE.GC_AUDITLOGS` - Performs `auditlogs` garbage collection

`STORAGEINITIATORPURGE.GC_STORAGEINITIATORS` - Performs `storage data` garbage collection

Manually Running GC Jobs

You can run GC jobs by logging in to SQL*Plus and entering the following:

```
# Su - oracle
# Sqlplus "/ as sysdba"

SQL> grant create session to gcadmin
SQL> connect gcadmin/<password>
SQL> exec dbms_job.run(<job no>);
```

For example, this sample command runs the `waypurge_gc` job:

```
SQL> exec dbms_job.run(4);
```

Monitor the ERROR_INTERNAL_MSG Table

The garbage collection jobs write exceptions to the `truth.ERROR_INTERNAL_MSG` table. You can monitor this table for errors (checking daily is recommended). For example:

```
# Su - oracle
# Sqlplus "/ as sysdba"

SQL> set line 200
SQL> col ERR_ID format 999999
SQL> col ERR_USER format a8
SQL> col ERR_TABLE format a25
SQL> col ERR_TABLE_PK_ID format a10
SQL> col ERR_CODE format 9999999
SQL> col ERR_TEXT format a20
SQL> col ERR_INFO format a30

SQL> select ERROR_INTERNAL_MSG_ID ERR_ID,
      ERR_DATE,
      ERR_USER,
      ERR_TABLE,
      ERR_TABLE_PK_ID,
      ERR_CODE,
      ERR_TEXT,
      DELETE_FLG,
      ERR_INFO
     from ERROR_INTERNAL_MSG
     order by ERR_DATE;
```

Database Monitoring Strategy

Because the Model Repository is a critical component of SA, the DBA should implement a monitoring strategy. The DBA can write custom monitoring scripts or use third-party products.

This section contains example commands for monitoring the Oracle database used by the Model Repository. When issuing the commands shown in this section, you must be logged on to the server as the user `oracle`:

```
$ su - oracle
```

The SQL commands shown in this section are entered in the `sqlplus` command-line utility. To run `sqlplus`, log on as `oracle` and enter the following command:

```
$ sqlplus "/ as sysdba"
```

Verify that the Database Instances are Up and Responding

To verify that the Database Instances are up and running, perform the following steps:

1. Check to see if the Oracle processes are running by entering the following command:

   ```
   ps -ef | grep ora_
   ```

   This `ps` command should generate output similar to the following lines:

   ```
   oracle    1883     1  0 Jul24 ?        00:00:00 ora_pmon_truth
   oracle    1885     1  0 Jul24 ?        00:00:00 ora_psp0_truth
   oracle    1887     1  0 Jul24 ?        00:00:00 ora_mman_truth
   oracle    1891     1  0 Jul24 ?        00:00:45 ora_dbw0_truth
   oracle    1895     1  0 Jul24 ?        00:01:11 ora_lgwr_truth
   oracle    1897     1  0 Jul24 ?        00:00:02 ora_ckpt_truth
   oracle    1899     1  0 Jul24 ?        00:00:24 ora_smon_truth
   oracle    1901     1  0 Jul24 ?        00:00:00 ora_reco_truth
   oracle    1903     1  0 Jul24 ?        00:00:02 ora_cjq0_truth
   oracle    2391     1  0 Jul24 ?        00:00:00 ora_qmnc_truth
   oracle    2513     1  0 Jul24 ?        00:00:00 ora_q000_truth
   oracle    2515     1  0 Jul24 ?        00:00:00 ora_q001_truth
   oracle   18837     1  0 03:04 ?        00:00:00 ora_mmon_truth
   oracle   18839     1  0 03:04 ?        00:00:00 ora_mmnl_truth
   oracle   25184 24635  0 21:35 pts/1    00:00:00 grep ora_
   ```

2. Verify that the database status is `ACTIVE` by entering the following command in `sqlplus`:

   ```
   select database_status from v$instance;
   ```

3. Verify that the open mode is `READ WRITE` by entering the following command in `sqlplus`:

   ```
   select name, log_mode, open_mode from v$database;
   ```

Verify that the Datafiles are Online

To verify that the datafiles are online, in `sqlplus`, enter the following commands:

```
Col file_name format a50
Col status format a10
Set line 200
Select file_id, status, bytes, file_name from dba_data_files order by
tablespace_name;
```

The status should be AVAILABLE for all the data files.

Verify That the Listener is Running

To verify that the listener is running, perform the following steps:

1. Check to see if the Oracle listener processes are running by entering the following command:

   ```
   ps -ef | grep tns

   oracle    1762     1  0 Jul24 ?        00:00:01 /u01/app/oracle/product/
   10.2.0/db_1/bin/tnslsnr LISTENER -inherit
   oracle   25231 25189  0 21:39 pts/1    00:00:00 grep tns
   ```

2. Check the status of the listener with the `lsnrctl` command:

   ```
   lsnrctl status
   ```

   The listener should be listening on port 1521 (default), or on the port that you have designated that the Oracle listener process use, with the TCP protocol, and should be handling the instance named truth. The `lsnrctl` command should generate output similar to the following lines:

   ```
   . . .
   Connecting to (ADDRESS=(PROTOCOL=tcp)
   (HOST=per1.performance.qa.example.com)(PORT=1521))
   . . .
   Instance "truth", status READY, has 1 handler(s) for this service...
   ```

3. Test connectivity to the instance from the Data Access Engine (spin) and Web Services Data Access Engine (twist) hosts by running the `tnsping` utility:

   ```
   tnsping truth
   ```

   The `OK` statement displayed by the `tnsping` utility confirms that the listener is up and can connect to the instance. The `tnsping` utility should generate output similar to the following lines:

   ```
   . . .
   Used parameter files:

   Used HOSTNAME adapter to resolve the alias
   Attempting to contact
   (DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=truth.performance.qa.example.com
   ))(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.165.178)(PORT=1521)))
   OK (0 msec)


   Attempting to contact
   (DESCRIPTION=(ADDRESS=(HOST=localhost)(PORT=1521)(PROTOCOL=tcp))(CONNECT_
   DATA=(SERVICE_NAME=truth)))
   OK (0 msec)
   ```

   As an alternative to running the `tnsping` utility in this step, you can check the connectivity by running `sqlplus` and connecting to the database instance with the service name (TNS alias), for example:

   ```
   sqlplus myuser/mypass@truth
   ```

Examine the Log Files

To examine the log files, perform the following steps:

1. Look for errors in the `alert.log` file.

   For each instance, locate the `alert.log` file in the background dump destination directory:

   Oracle 10g

   `$ORACLE_BASE/admin/<SID>/bdump`

   Oracle 11g

   `$ORACLE_BASE/diag/rdbms/<SID>/<SID>/trace/`

   Here is an example `bdump` directory for an instance with the `truth` SID:

   Oracle 10g

   `/u01/app/oracle/admin/truth/bdump`

   Oracle 11g

   `/u01/app/oracle/diag/rdbms/truth/truth/trace/`

2. Look for errors in the other log and trace files, located in the following directories:

   Oracle 10g

   ```
   $ORACLE_BASE/admin/<SID>/cdump
   $ORACLE_BASE/admin/<SID>/adump
   $ORACLE_BASE/admin/<SID>/udump
   ```

   Oracle 11g

   Various direcrories under:

   `$ORACLE_BASE/diag/rdbms/<SID>/<SID>`

Check for Sufficient Free Disk Space in the Tablespaces

To check for sufficient disk space, perform the following steps:

1. Enter the following commands in `sqlplus`:

```
column dummy noprint
column  pct_used format 999.9       heading "Pct|Used"
column  name    format a16      heading "Tablespace Name"
column  Kbytes   format 999,999,999    heading "Current|File Size|MB"
column  used    format 999,999,999   heading "Used MB "
column  free    format 999,999,999  heading "Free MB"
column  largest    format 999,999,999  heading "Largest|Contigous|MB"
column  max_size format 999,999,999 heading "Max Possible|MB"
column  pct_max_used format 999.999       heading "Pct|Max|Used"
break   on report
compute sum of kbytes on report
compute sum of free on report
compute sum of used on report

select nvl(b.tablespace_name,
             nvl(a.tablespace_name,'UNKOWN')) name,
       kbytes_alloc Kbytes,
```

```
       kbytes_alloc-nvl(kbytes_free,0) used,
       nvl(kbytes_free,0) free,
       ((kbytes_alloc-nvl(kbytes_free,0))/
                        kbytes_alloc)*100 pct_used,
       nvl(largest,0) largest,
       nvl(kbytes_max,kbytes_alloc) Max_Size,
       ((kbytes_alloc-nvl(kbytes_free,0))/kbytes_max)*100 pct_max_used
from ( select sum(bytes)/1024/1024 Kbytes_free,
              max(bytes)/1024/1024 largest,
              tablespace_name
       from  sys.dba_free_space
       group by tablespace_name ) a,
     ( select sum(bytes)/1024/1024 Kbytes_alloc,
              sum(decode(maxbytes,0,bytes,maxbytes))/1024/1024 Kbytes_max,
              tablespace_name
       from sys.dba_data_files
       group by tablespace_name
       union all
      select sum(bytes)/1024/1024 Kbytes_alloc,
              sum(decode(maxbytes,0,bytes,maxbytes))/1024/1024 Kbytes_max,
              tablespace_name
       from sys.dba_temp_files
       group by tablespace_name) b
where a.tablespace_name (+) = b.tablespace_name
order by 1
/
```

In the output generated by the preceding commands, compare the numbers under the `Used` and `Free` headings.

2 To list the existing data, index, and temporary files, enter the following commands in `sqlplus`:

```
Select file_id, bytes, file_name from dba_data_files;
Select file_id, bytes, file_name from dba_temp_files;
```

3 If a tablespace has auto-extended to its maximum size and is running out of disk space, then add new data files by entering the `ALTER TABLESPACE` command in `sqlplus`.

The following example commands add data files to four of the tablespaces. For a full list of tablespaces and data files, see the output generated by the commands in the preceding two steps.

```
ALTER TABLESPACE "AAA_DATA"
ADD DATAFILE '/u01/oradata/truth/aaa_data10.dbf'
SIZE 32M AUTOEXTEND ON NEXT 128M MAXSIZE 4000M ;

ALTER TABLESPACE "AAA_INDX"
ADD DATAFILE '/u02/oradata/truth/aaa_indx11.dbf'
SIZE 32M AUTOEXTEND ON NEXT 128M MAXSIZE 4000M ;

ALTER TABLESPACE "UNDO"
ADD DATAFILE '/u03/oradata/truth/undo12.dbf' SIZE 32M AUTOEXTEND ON NEXT
128M MAXSIZE 4000M ;

ALTER TABLESPACE "TEMP" ADD
TEMPFILE '/u04/oradata/truth/temp14.dbf' SIZE 32M AUTOEXTEND ON NEXT 128M
MAXSIZE 4000M ;
```

Verify that the Database Jobs (System/Schema Statistics and Garbage Collection) Ran Successfully

When the Model Repository is installed, the SA Installer sets up the System/Schema Statistics and the Garbage Collection jobs in Oracle's dba_jobs. dba_jobs runs these jobs at specified time-intevals. The jobs perform system/schema statistics collection and garbage collection. If the system/schema statistics collection jobs do not run successfully, database performance will degrade. If the garbage collection jobs do not run, then old data will accumulate requiring additional disk space. Performance can also be affected.

To verify that the Jobs in DBA_JOBS ran successfully, perform the following steps:

1. Enter the following commands in SQL*Plus:

```
# Su - oracle
# Sqlplus "/ as sysdba"
SQL> set line 200
SQL> col priv_user format a14
SQL> col last format a17
SQL> col next format a17
SQL> col this format a17
SQL> col what format a50
SQL> col broken format a1

SQL> select job, priv_user, to_char(LAST_DATE, 'MM/DD/YY HH:MI:SS') last,
to_char(NEXT_DATE, 'MM/DD/YY HH:MI:SS') next, broken, what from dba_jobs;
```

   In the output generated from the preceding statement, the value of the WHAT column indicates the type of job. If the value of WHAT is DBMS_STATS* or GATHER_*, the job performs statistics collection. The jobs owned by GCADMIN perform the garbage collection. Sample output looks like this:

```
 JOB PRIV_USER        LAST              NEXT              B WHAT
---- -------------- ----------------- ----------------- - --------------------------------------------------
  21 TRUTH          05/04/09 11:00:03 05/06/09 11:00:00 N DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'TRUTH', e
                                                          stimate_percent=>dbms_stats.auto_sample_size, degr
                                                          ee=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO', o
                                                          ptions=>'GATHER', cascade=>TRUE, gather_temp=>TRUE
                                                          );

  22 AAA            05/04/09 11:00:03 05/06/09 11:00:00 N DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA', est
                                                          imate_percent=>dbms_stats.auto_sample_size, degree
                                                          =>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO', opt
                                                          ions=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);
  23 LCREP          05/04/09 11:00:03 05/06/09 11:00:00 N gather_lcrep_stats;
  24 GCADMIN        05/05/09 09:00:04 05/06/09 09:00:00 N WAYPURGE.GC_SESSIONS;
  25 GCADMIN        05/05/09 09:00:04 05/06/09 09:00:00 N CHANGELOGPURGE.GC_CHANGELOGS;
  26 GCADMIN        05/05/09 09:00:04 05/06/09 09:00:00 N AUDITPURGE.GC_AUDITLOGS;
  27 GCADMIN        05/05/09 05:25:08 05/05/09 06:25:08 N STORAGEINITIATORPURGE.GC_STORAGEINITIATORS;
  28 OPSWARE_ADMIN  05/04/09 06:00:02 05/11/09 06:00:00 N DBMS_STATS.GATHER_SYSTEM_STATS(gathering_mode=>'IN
                                                          TERVAL', interval=>30);


8 rows selected.
```

   where:

   JOB - job id

   SCHEMA_USER - the user who with permissions to run the job

   LAST_DATE - last date-time when the job was run

   NEXT_DATE - next date the job will run

   BROKEN - value N = job was successful, value = Y - job failed

   WHAT - the type of job

Changes to the Database Statistics Job

The following changes have been made to the database statistics collection jobs. These jobs can be found in the `dba_jobs` table. These changes are only relevant to upgraded SA Cores.

To view the jobs you can run the following from SQL*plus

```
# Su - oracle
# Sqlplus "/ as sysdba"
set line 200
col priv_user format a14
col what format a50
col job format 999
select job, priv_user, what from dba_jobs where priv_user in ('AAA','TRUTH');
```

Your output should be as follows:

SA 7.50:

```
## TRUTH DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'TRUTH', options=>'GATHER AUTO');
## AAA   DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA', options=>'GATHER AUTO');
```

SA 7.80:

```
 ## TRUTH DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'TRUTH',
          estimate_percent=>dbms_stats.auto_sample_size,
          degree=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
          options=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);
 ## AAA   DBMS_STATS.GATHER_SCHEMA_STATS(ownname=>'AAA',
          estimate_percent=>dbms_stats.auto_sample_size,
          degree=>10, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
          options=>'GATHER', cascade=>TRUE, gather_temp=>TRUE);
```

Backups of the dba_jobs.what information Table

During the SA 7.80 Model Repository Guide upgrade, the SA 7.50 `dba_jobs.what` information table is backed up and then replaced by the SA 7.8 `dba_jobs.what` table. You can view the backed up information by logging in to SQL*Plus and entering the following commands:

```
# Su - oracle
# Sqlplus "/ as sysdba"
SQL> set line 200
SQL> col ERR_ID format 999999
SQL> col ERR_USER format a8
SQL> col ERR_TABLE format a10
SQL> col ERR_TABLE_PK_ID format a10
SQL> col ERR_CODE format 9999999
SQL> col ERR_TEXT format a20
SQL> col ERR_INFO format a30

SQL> select ERROR_INTERNAL_MSG_ID ERR_ID,
SQL>  ERR_DATE,
SQL>  ERR_USER,
SQL>  ERR_TABLE,
SQL>  ERR_TEXT,
SQL>  ERR_INFO
SQL> from ERROR_INTERNAL_MSG where ERR_TEXT = 'SA7.8 Model Repository Upgrade'
order by ERR_DATE;
```

Output will look similar to the following:

```
 ERR_ID ERR_DATE  ERR_USER ERR_TABLE  ERR_TEXT               ERR_INFO
------- --------- -------- ---------- --------------------   ------------------------------
      6 07-MAY-09 TRUTH    DBA_JOBS   SA7.8 Model Repository Pre SA7.8 dba_jobs.what value
                                      Upgrade               was: DBMS_STATS.GATHER_SCHEMA_
                                                            STATS(ownname=>'TRUTH', options
                                                            =>'GATHER AUTO');

      5 07-MAY-09 AAA      DBA_JOBS   SA7.8 Model Repositryo Pre SA7.8 dba_jobs.what value
                                      Upgrade               was: DBMS_STATS.GATHER_SCHEMA_
                                                            STATS(ownname=>'AAA', options=
                                                            >'GATHER AUTO');
```

Running the dba_jobs manually

If you need to run the System/Schema Statistics and the Garbage Collection jobs manually, you must first grant the following privilege.

```
SQL> grant create session to truth, aaa, lcrep;
```

To run the statistics collection jobs manually in SQL*Plus, use the commands shown below.

If you copy and paste the following command examples, replace the variables like `schema_user_value` with the values of the `schema_user` column displayed by the preceding select statement. Substitute the variables such as `job_no_value` with the values of the `job` column displayed by the same select statement.

```
SQL> connect <schema_user_value>/<password>
SQL> exec dbms_job.run(<job_no_value>)
```

After you are done running the jobs, you should revoke the privileges granted above. Log in to SQL*Plus and enter the following command:

```
SQL> revoke create session from truth, aaa, lcrep;
```

Monitor Database Users

To monitor database users, perform the following steps:

1. To check the database users, enter the following command in `sqlplus`:

   ```
   Select username, account_status, default_tablespace,
   temporary_tablespace from dba_users;
   ```

Troubleshooting System Diagnosis Errors

If an additional privilege (permission) has been made manually to the database, when SA performs a system diagnosis on the Data Access Engine, an error message might be generated. For example, if an additional grant has been made to the `truth.facilities` table, the following error appears:

```
Test Information
Test Name: Model Repository Schema
Description: Verifies that the Data Access Engine's version of the schema
matches the Model Repository's version.
Component device: Data Access Engine (spin.blue.qa.example.com)
Test Results: The following tables differ between the Data Access Engine
and
the Model Repository:  facilities.
```

To fix this problem, revoke the grant. For example, if you need to revoke a grant on the `truth.facilities` table, log on to the server with the database and enter the following commands:

```
su - oracle
sqlplus "/ as sysdba"
grant create session to truth;
connect truth/<truth passwd>;
revoke select on truth.facilities from spin;
exit
sqlplus "/ as sysdba"
revoke create session from truth;
```

Oracle Database Backup Methods

It is important that you back up the database on a regular basis. Be sure to use more than one backup method and to test your recovery process.

You can use the following methods to back up the Oracle database:

- **Export-Import**: An export extracts logical definitions and data from the database and writes the information to a file. Export-import does not support point-in-time recoveries. Do not use Export-Import as your only backup and recovery strategy.

  See the information on the `Export-Import` subdirectory in Oracle/SA Installation Scripts, SQL Scripts, and Configuration Files on page 201.

- **Cold or Off-Line Backups**: This procedure shuts the database down and backs up all data, index, log, and control files. Cold or off-line backups do not support point-in-time recoveries.
- **Hot or Online Backups**: During these backups, the database must be available and in `ARCHIVELOG` mode. The tablespaces are set to backup mode. This procedure backs up tablespace files, control files, and archived redo log files. Hot or online backups support point-in-time recoveries.
- **RMAN Backups**: While the database is either off-line or on-line, use the `rman` utility to back up the database.

Regardless of your backup strategy, remember to back up all required Oracle software libraries, parameter files, password files, and so forth. If your database is in `ARCHIVELOG` mode, you also need to back up the archived log files.

For more information on backing up Oracle databases, see the following Oracle documents:

- *Oracle Database 2 Day DBA*
- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*

These guides are on the Oracle web site at the following URL:

```
http://www.oracle.com/technology/documentation/index.html
```

Useful SQL

The following SQL commands help you manage information in the Oracle database that the Model Repository uses.

Locked and Unlocked User

A user in Oracle 10.2.0.2 will be locked out after ten unsuccessful log on attempts.

To verify whether the user has been locked or unlocked, enter the following SQL command:

```
select username, account_status from dba_users;
```

To unlock the user, enter the following sql command:

```
>ALTER USER <username> ACCOUNT UNLOCK;
```

GATHER_SYSTEM_STATS

Sometimes the GATHER_SYSTEM_STATS job will be suspended. To remove this from 'AUTOGATHERING" mode, perform the following steps:

1. Select PNAME, pval2 from SYS.AUX_STATS$ where pname ='STATUS';.
2. If the PVAL2 status is "AUTOGATHERING", run GATHER_SYSTEM_STATS with gathering_mode=('STOP');.
3. Run your job 'exec dbms_job.run(xxx);.

BIN$ Objects

If the SA Installer discovers the existence of BIN$ objects in the database, enter the following sql commands:

```
show parameter recyclebin;

SELECT owner,original_name,operation,type FROM dba_recyclebin;
connect <owner>/password
purge recyclebin; or purge table BIN$xxx;
```

By default, recyclebin is set to OFF.

B SA Gateway Properties File

This section provides reference information about the parameters in the Gateway Properties file used by the SA Gateway.

SA Gateway Properties File Syntax

An SA Gateway properties file can have the following entries:

Usage: `./opswgw-tc-70 [options]`

`--Gateway name`

(**Required**) Set the name of the SA Gateway. This name must be unique in a Gateway mesh.

`--Realm realm`

(**Required**) All Gateways operate in a named Realm. A *Realm* is an SA construct that refers to a set of servers which are serviced by the Gateways in the Realm. Realms can support an IPV4 address space which may overlap with other Realms. Realms are also used to define bandwidth utilization constraints on SA functions.

`--Root true | false`

Specifies that this Gateway will act as a root of the Gateway mesh. All Gateways in a Root Realm must be Root Gateways.

Default: false.

`--Level int`

(*Experimental*) Routing level for the Gateway. There are eight possible levels, 0-7. All Gateways in a realm must have the same level.

Default: 0

`--GWAddress lhost`

Sets the local host address (if you are specifying the value for the Management Gateway, use the IP address only, do not use the hostname; you can, however, use the hostname for other, non-Management Gateways) that this Gateway uses to tell other components how to contact it. This value is used by the core to discover new core-side Gateways. It is also used to communicate the active list of Gateways that are servicing Realm to proxy clients (such as Agents) via the `X-OPSW-GWLIST` mime header.

`--Daemon true | false`

Daemonize the process.

Default: false.

`--Watchdog true | false`

Start an internal watchdog process to restart the Gateway in case of a failure or signal. A `SIGTERM` sent to the watchdog will stop the watchdog and Gateway processes.

Default: false.

`--User name`

Change to this user on startup.

`--RunDir path`

Change to this directory on startup.

`--ChangeRoot true | false`

If `true` chroot into `RunDir`. This can to used by a helper script to construct a jail.

Default: false

`--PreBind proto:ip:port, ...`

For security reasons, it can be useful to run a Gateway chrooted as a non-privileged use (only ports above 1024 can be used for any listeners). If you want to use a non-privileged user *and* a privileged listener port, you can use the `--PreBind` directive to reserve the port while the process is `root` and before privileges are dropped.

`--HardExitTimeout seconds`

The number of seconds after a restart or exit request that the main thread will wait for internal threads and queues to quiesce before performing a hard exit.

`--LogLevel INFO | DEBUG | TRACE`

Sets the logging level. Note that `DEBUG` and `TRACE` can produce a large amount of output, which is typically relevant only to developers, and can negatively affect performance.

Default: `INFO`.

`--LogFile file`

The filename of the SA log file.

`--LogNum num`

The number of rolling log files to keep.

`--LogSize size`

The size in bytes of each log file.

`--TunnelDst [lip1:]lport1[:crypto1],...`

If specified, starts a tunnel destination listener. The tunnel listener can listen on multiple ports (a comma-separated list with no spaces). If the port is prefixed with an IP address, the listener will bind only to that IP address. For example: `2001, 10.0.0.2:2001, 2001:/var/foo.pem, 10.0.0.2:2001:/var/foo.pem`

`--TunnelSrc rhost1:rport1:cost1:bw1[:crypto1],...`

If specified, creates a tunnel between this Gateway and the Gateway listening at `rhost1:rport1`. The link `cost1` and link bandwidth `bw1` must be set. The cost is a 32-bit unsigned int, and bandwidth is in Kbits/sec (K=1024bits). (Additional tunnels are separated by commas.) Examples: `gw.foo.com:2001:1:0, gw.bar.com:2001:10:256:/var/foo.pem`

`--ProxyPort [lip1:]lport1,[lip2:]lport2,...`

The HTTP CONNECT proxy listener port. If more than one proxy listener port is needed, you can add more using a comma separated list. You can enable interface binding by prepending an IP address to the port.

`--ForwardTCP [lip1:]lport1:realm1:rhost1:rport1,...`

Creates a static TCP port forward. Forward the local port `lport(`x`)` to the remote service `rhost(`x`):rport(`x`)`, which is in `realm(`x`)`. A blank `realm` (such as `lport::rhost:rport`) means route to the closest Root Realm.

`--ForwardTLS  [lip1:]lport1:realm1:rhost1:rport1, ...`

Creates a static TCP port forward that specializes in TLS traffic. The TLS session ID is parsed and sent to the egress Gateway for use in load balancing algorithms. In all other respects, this feature behaves like `ForwardTCP`.

`--ForwardUDP [lip1:]lport1:realm1:rhost1:rport1,...`

Creates a static UDP port forward. Forward local port `lport(`x`)` to remote service `rhost(`x`):rport(`x`),` which is in `realm(`x`)`. A blank `realm` (such as `lport::rhost:rport`) means route to the closest Root Realm. (Note: Some UDP services, such as DHCP, cannot be proxied in this way.)

`--IdentPort [lip:]lport`

Starts an IDENT service listening on local port `lport` (optionally bound to the local IP `lip`.

`--AdminPort [lip:]lport[:crypto1]`

Starts an administration interface listening on local port `lport`, which is optionally bound to the local IP `lip`. If you use `crypto`, include a `crypto` specification file name.

`--ConnectionLimit int`

Specifies the soft memory tuning limit for the ‘maximum number of connections.

`--OpenTimeout seconds`

Wait a maximum `seconds` for a remote `CONNECT` call to establish a remote connection.

`--ConnectTimeout seconds`

Wait a maximum `seconds` for a `connect()` to complete. If a timeout occurs, then an `HTTP 503` message is returned to the client (via the ingress Gateway). The client will get this message if the `ConnectTimeout` plus the Gateway mesh transit delay is less than the `OpenTimeout`.

`--ReorderTimeout seconds`

In the event of out-of-order messages (for a TCP flow), limits the amount of time (`seconds`) to wait for messages needed for reassembly to arrive. The most common cause of out-of-order messages is when a transit tunnel fails and a new route is taken mid-flow.

`--TunnelStreamPacketTimeout seconds`

If a portion of a TCP flow cannot be delivered to an endpoint, then teardown the TCP connection after `seconds`.

`--QueueWaitTimeout seconds`

Specifies the maximum time that a tunnel message can wait at the head of an internal routing queue (while waiting for a tunnel to be restored).

`--KeepAliveRate seconds`

Send link `keepalive` messages once every *x* seconds on each link.

`--LsaPublishRateMultiple float`

Link State Advertisements (LSAs) are published once every `k*M` seconds. Where `M` is the number of Gateways in the mesh and `k` is a floating point constant specified using `--LsaPublishRateMultiple`. For example, if there are 100 Gateways in a mesh and `--LsaPublishRateMultiple` is set to 2.0, then an LSA is published approximately every 200 seconds (due to implementation factors, the actual delay will be somewhere between 190 and 210 seconds).

`--LsaTTLMultiple float`

Sets the TTL for LSAs to `float` multiplied by the `LsaPublishRate`. Example: If `LsaPublishRate` is 10 seconds and `LsaTTLMultiple` is 3 then, the TTL for LSAs published by this Gateway is set to 30 seconds.

`--MaxRouteAge seconds`

Discards the routes from the routing table that have not been refreshed within `seconds`.

`--RouteRecalcDutyCycle  percentage`

If the time to calculate Dijkstra takes `tau` seconds, then wait for `tau*(1/RouteRecalcDutyCycle-1)` seconds until another recalculation can take place.

`--TunnelTimeoutMultiple float`

This number, multiplied by the `KeepAliveRate`, gives the maximum time that a tunnel can be idle before it is garbage collected.

`--DoNotRouteService host1:port1,host2:port2,...`

Specifies that, when a local client creates a proxy connection to `host:port`, do not route the message; service it locally. Use this property to ensure that certain services are handled locally, in the Gateway's current Realm.

`--ForceRouteService host1:port1:realm1,host2:port2:realm2,...`

When a local client creates a proxy connection to `host:port`, force the message to route to a specified Realm.

`--HijackService host1:port1,host2:port2,...`

When the local Gateway sees a connection to `host:port` via a tunnel, and the source Realm is not the local Realm, it must service the connection. If the connection is from the local Realm, the Gateway must allow the message to continue to its destination. You can use this feature to implement transparent caches.

`--RouteMessages  *true | false`

If specified as `true,` turn on transit routing. If `false`, disable transit routing. If the destination of the message is *not* the local Gateway, then, by default, the message is routed based on the current routing table. If such routing is not desired set this property to `false`.

`--EgressFilter proto:dsthost1:dstport1:srchost1:srcrealm1,...`

When the local Gateway sees a `TCP` connection attempt to `dsthost:dstport` from `srchost1:srcrealm1`, it must allow the connection. The implied default is to deny all connections. If you want to *allow* all connections, specify the egress filter as `*:*:*:*:*`. It is also common for an egress filter to only allow connections from the Root Realm. This can be expressed by leaving the `srcrealm` blank. Example: `tcp:10.0.0.5:22:172.16.0.5:` would allow tcp connections to 10.0.0.5, port 22, from 172.16.0.5 in a Root Realm.

`--IngressMap ip1:name,ip2:name,...`

When sending an open message (and the `srcip` is in the ingress map), append (as metadata) the `ip:name` mapping to the open message. This allows a remote egress filter to use the `name` as the `srchost` instead of the `ip`. This feature supports the addition of a server to a farm without the need to individually add the server to many `EgressFilter` entries.

`--LoadBalanceRule proto:thost:tport:mode:rhost1:rport1:`
`rhost2:rport2, ...`

When receiving a new connection message for `thost:tport`, load balance the connection over real hosts `rhost1:rport1, rhost2:rport2` etc. The load balance strategy is defined by `mode`.

There are six load-balancing modes:

STICKY: Send the connection to a working target based on a priority list randomized by a hash of the source IP and source Realm (the hash string can be overridden via the input MIME header X-OPSW-LBSOURCE).

LC: Send connection to a working target with the least number of connections.

RR: Send connection to the next working target in a round-robin fashion.

TLS_STICKY: Use an `SSLv3/TLSv1.0` session ID to send the connection back to the previous target based on a session ID cache. If the target is in error, or the session ID is missing from the cache, fall back to STICKY mode to make a new selection.

TLS_LC: Similar to TLS_STICKY mode, but falls back to LC mode (least connections).

TLS_RR: Similar to TLS_STICKY mode, but falls back to RR mode (round-robin). Remember to add an egress filter for `proto:thost:tport`. You do not need to add egress filters for the targets. Non-TLS load balancing modes *can* be used with UDP services.

`--LoadBalanceRetryWindow seconds`

If an error occurs when using a load balanced target (such as `rhost1:rport1` above) then the target is marked `in-error`. This property controls how many seconds a Gateway will wait until it re-tries the target. If the target is missing (such as an `RST` is received upon the connection request) the load balancer will silently try to find a good target.

`--SessionIdTimeout  seconds`

The number of seconds a load balanced `SSLv3/TLS` client can be idle before the `sessionId` association is reaped. This property affects the egress Gateway of a TLS flow.

`--SessionIdCacheLimit  slots`

A soft limit on the number of `SSLv3/TLS` session IDs that the cache can hold. If this limit is exceeded, then the garbage collector begins reducing the `SessionIdTimeout` value in order to achieve the cache limit specified by `--SessionIdCacheLimit`.

`--MinIdleTime seconds`

Specifies the minimum number of `seconds` a connection can be idle during an overload condition before it will be considered for reaping.

`--GCOverloadTrigger float`

Specifies the fraction of `SoftConnectionLimit` at which to start overload protection measures. When the number of open connections hits this overload trigger point, overload protection starts, reaping the most idle connections over `MinIdleTime`. Overload protection stops when the connection count falls below the overload trigger point.

`--GCCloseOverload true | false`

When a client tries to open a connection after the `ConnectionLimit` has been reached, this property tells the Gateway what to do with the new connection. A value of `true` causes the Gateway to close the new connection. A value of `false` causes the Gateway to park the new connection in the kernel's backlog and to service it after the overload condition subsides. The proper setting is application dependent.

Default: `false`.

`--VerifyRate seconds`

When a connection stops moving data for the specified number of `seconds`, a connection verify message is sent to the remote Gateway to verify that the connection is still open. This check is repeated periodically and indefinitely when the timeout has expired.

`--OutputQueueSize slots`

Specifies the size of the tunnel output queues. These queues store messages destined for remote Gateways. Each remote Gateway has an output queue. Queues are garbage collected after `MaxQueueIdleTime` is reached.

`--MaxQueueIdleTime seconds`

Specifies the maximum time to keep an idle output queue before garbage collection removes it.

`--TunnelManagementQueueSize  slots`

Specifies the size of the queues used to manage tunnel management traffic, such as Link State Advertisements.

`--TunnelTCPBuffer bytes`

Specifies the size of the `TCP SEND` and `RECV` buffer in bytes. The operating system must be configured to handle the specified value. You can view the Gateway's log file to see if the specified is denied by the operating system.

`--DefaultChunkSize bytes`

Specifies the default (maximum) IO chunk size when encapsulating a TCP stream. This property value can be applied only to links with no bandwidth constraint.

`--LinkSaturationTime seconds`

When a links has a bandwidth constraint, the chunk size, `DefaultChunkSize`, is computed based on two parameters. The first is the link's bandwidth constraint. The second is the amount of time that the bandwidth shaper should utilize the full, real, bandwidth on the link. This parameter controls the duty cycle of the bandwidth shaper. Smaller values give a smoother bandwidth control at the cost of more overhead, because each smaller IO chunk has a header.

`--TunnelPreLoad slots`

Specifies the maximum number of output queue slots to use before waiting for the first `Ack` message. This allows for pipelining in Long Fat Pipes. This value is reduced geometrically to one as the number of queue s'lots diminish.

`--BandwidthAveWindow samples`

Specifies the maximum number of IO rate samples for the bandwidth estimation moving window. The samples in this window are averaged to provide a low pass estimate of the bandwidth in use by a tunnel. This estimate has high frequency components due to the sharp edge of the filter window.

`--BandwidthFilterPole float`

Specifies the pole of a discrete-time first-order smoothing filter used to remove the high frequency components of the moving window estimator. Set the value to 0.0 to turn off this filter.

`--StyleSheet URL`

Adds a stylesheet link to a URL when rendering the admin UI. This is useful for embedding the admin UI in another web-based UI. In addition to using this property to control the default stylesheet, a dynamic stylesheet override is supported by adding the variable `StyleSheet=<url>/style.css` to the admin UI URL.

`--ValidatePeerCN true | false`

Specifies whether the peer CN is validated against the peer configuration during a tunnel handshake operation. The peer must be turned off during the installation of an untrusted Gateway.

Default: true.

`--PropertiesCache file`

Link cost and bandwidth can be controlled via `parameter-modify` messages over tunnel connections. These real-time adjustments are made to the running process and written to a parameter cache which will override the properties file or command-line arguments.

`--PropertiesInclude file`

Specifies an Include file to load and merge with the current properties. Properties in the include file can override properties from the original Properties File. This property can be specified from the command line. If so, it will override *all* properties, including command line overrides. It is not recursive and does not support a list.

`--PropertiesFile  file`

Places all command-line arguments into a properties file within the `opswgw` name space. Note that, the `PropertiesFile` command-line argument itself *must not* be placed in the properties file within the `opswgw` name space.

opswgw Command-Line Arguments

All of the parameters in the preceding section can be specified as options for the `opswgw` command. For example, the `opswgw.Gateway foo` entry in the Gateway Properties file is equivalent to the following command-line argument:

```
/opt/opsware/opswgw/bin/opswgw --Gateway foo
```

Command-line arguments override corresponding entries in the Gateway Properties file. In addition to the entries listed in the preceding section, the `opswgw` command can specify a Gateway Properties file as an argument, for example:

```
/opt/opsware/opswgw/bin/opswgw --PropertiesFile filename
```

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W