



Opsware[®] SAS 5 Administration Guide

Corporate Headquarters

599 North Mathilda Avenue Sunnyvale, California 94085 U.S.A.
T + 1 408.744.7300 F +1 408.744.7383 www.opsware.com

Opware SAS Version 5.5.1

Copyright © 2000-2006 Opware Inc. All Rights Reserved.

Opware Inc. Unpublished Confidential Information. NOT for Redistribution. All Rights Reserved.

Opware is protected by U.S. Patent Nos. 6,658,426, 6,751,702, 6,816,897, 6,763,361 and patents pending

Opware, Opware Command Center, Model Repository, Data Access Engine, Web Services Data Access Engine, Software Repository, Command Engine, Opware Agent, Model Repository Multimaster Component, and Code Deployment & Rollback are trademarks and service marks of Opware Inc. All other marks mentioned in this document are the property of their respective owners.

Additional proprietary information about third party and open source materials can be found at <http://www.opware.com/support/opensource.doc.pdf>.

Table of Contents

Preface	11
<hr/>	
Overview of this Guide	11
Contents of this Guide	11
Conventions in this Guide	12
Icons in this Guide	13
Guides in the Documentation Set and Associated Users	14
Opsware, Inc. Contact Information	14
Chapter 1: Opsware SAS Overview	17
<hr/>	
Opsware SAS Technology	17
Types of Opsware Users	19
Opsware SAS Environment	20
Model-Based Control	21
Types of Opsware SAS Installations	22
Opsware SAS Components	23

Boot Server26
Build Manager26
Command Engine26
Data Access Engine26
Media Server26
Model Repository27
Model Repository Multimaster Component27
Opsware Agents27
Dormant Opsware Agents28
Opsware Command Center28
OS Build Agent29
Software Repository29
Software Repository Replicator29
Software Repository Cache30
Software Repository Multimaster Component30
Web Services Data Access Engine30
Opsware Gateway30
Global File System Server31
Interaction Among Opsware SAS Components31

General Interaction Among Components	31
Opware SAS Security	32
OS Provisioning	32
Patch Management	36
Software Provisioning	43
Code Deployment and Rollback	46
Script Execution	48
Integration with AIX and HP-UX Installation Technology	51
Component Interaction in Multiple Facilities	53
Discovery and Agent Deployment	55
Application Configuration Management	57
Visual Packager	59
Server Compliance and Auditing	61

Chapter 2: Opware Multimaster Mesh Administration 65

Overview of Opware Multimaster Mesh	65
Multimaster Facilities Administration	66
Updating Facility Information and Settings	66
Multimaster Mesh Administration	68
Overview of Multimaster Mesh Administration	69
Model Repository Multimaster Component Conflicts	69
Causes of Conflicts	70
User Overlap	70
User Duplication of Actions	71
Connectivity Problems that Cause Out of Order Transactions	72
Best Practices for Preventing Multimaster Conflicts	73
Examining the State of the Multimaster Mesh	74

Best Practices for Resolving Database Conflicts	74
Types of Conflicts.....	75
Guidelines for Resolving Each Type of Conflict	75
Model Repository Multimaster Component Conflicts	78
Overview of Resolving Model Repository Multimaster Component Conflicts.....	78
Resolving a Conflict by Object.....	79
Resolving a Conflict by Transaction	84
Network Administration for Multimaster	88
Multimaster Alert Emails	88
Chapter 3: Opsware Satellite Administration	93
Overview of the Opsware Satellite	93
Opsware Gateway	95
Facilities and Realms.....	95
Satellite Information and Access	96
Permissions Required for Managing Satellites.....	96
Viewing Facilities	97
Enabling the Display of Realm Information	99
Viewing the Realm of a Managed Server.....	99
Viewing Gateway Information	101
Software Repository Cache Management	105

Availability of Packages on the Software Repository Cache.....	106
Ways to Distribute Packages to Satellites.....	107
Setting the Update Policy.....	110
On-demand Updates.....	111
Manual Updates.....	111
Hierarchical Software Repository Caches.....	112
Cache Size Management.....	112
Creation of Manual Updates.....	113
Creating a Manual Update Using the DCML Exchange Tool (DET) ...	113
Applying a Manual Update to a Software Repository Cache.....	115
Staging Files to a Software Repository Cache.....	116
Microsoft Utility Uploads and Manual Updates.....	117
Chapter 4: Opware SAS Maintenance	119
Possible Opware SAS Problems.....	119
Opware Component Troubleshooting.....	120
Contacting Opware Support.....	120
Opware SAS Diagnosis.....	121

Opsware SAS Diagnosis Tool Functionality	121
System Diagnosis Testing Process	122
System Diagnosis Test Components	122
Data Access Engine Tests	123
Software Repository Tests	124
Web Services Data Access Tests	124
Command Engine Tests	125
Model Repository Multimaster Component Tests	125
Running a System Diagnosis of Opsware Components	126
Logs for Opsware Components	127
Boot Server Logs	127
Build Manager Logs	128
Command Engine Logs	128
Data Access Engine Logs	128
Media Server Logs	128
Model Repository Logs	128
Model Repository Multimaster Component Logs	128
Opsware Agents Logs	129
Opsware Command Center Logs	129
Software Repository Logs	129
Software Repository Replicator Logs	129
Software Repository Multimaster Component Logs	129
Web Services Data Access Engine Logs	130
Opsware Gateway Logs	130
Global File System Server Logs	131
Global Shell Audit Logs	131

Shell Event Logs	131
Shell Stream Logs	133
Shell Script Logs	133
Digital Signatures in the Global Shell Audit Logs	133
Storage Management for the Global Shell Audit Logs	134
Configuring the Global Shell Audit Logs	136
Overview of Restarting Opsware Components	136
Restarting Opsware Components	137
Restarting the Boot Server	138
Restarting the Build Manager	138
Restarting the Command Engine	138
Restarting the Data Access Engine	139
Restarting the Media Server	139
Restarting the Model Repository	140
Restarting the Model Repository Multimaster Component	140
Restarting an Opsware Agent	141
Restarting the Opsware Command Center	142
Restarting the Software Repository	142
Restarting the Software Repository, Multimaster Component	142
Restarting the Web Services Data Access Engine	143
Restarting the Opsware Gateway	143
Restarting the Global File System Server	143
Opsware Software	144
Mass Deletion of Backup Files	145
Command Syntax	145
Deleting Backup Files with the Mass Deletion Script	145
Designations for Multiple Data Access Engines	148

Overview of Designations for Multiple Data Access Engines	148
Reassigning the Data Access Engine to a Secondary Role	149
Designating the Multimaster Central Data Access Engine	150
Web Services Data Access Engine Configuration File	151
Adding Locales to the Opware Command Center Component	153
Automatically Importing Windows Patches	154
Index	157

Preface

Welcome to the Opsware Server Automation System (SAS) – an enterprise-class software solution that enables customers to get all the benefits of the Opsware data center automation platform and support services. Opsware SAS provides a core foundation for automating formerly manual tasks associated with the deployment, support, and growth of server and server application infrastructure.

Overview of this Guide

This guide describes how to administer Opsware SAS, including how to administer multimaster facilities and administer Opsware Satellite. It also discusses how to monitor and diagnose the health of Opsware SAS components.

This guide is intended for Opsware administrators who will update facility information, resolve database conflicts in multiple core environments, manage the Software Repository Cache, monitor logs, and stop and restart components.

Contents of this Guide

This guide contains the following chapters:

Chapter 1: Opsware SAS Overview: Provides an overview and diagrams of Opsware SAS architecture, showing how Opsware SAS components and features interact both in single core and multiple core environments. Each of the components and its function is introduced.

Chapter 2: Opsware Multimaster Mesh Administration: Provides information about how to manage data across facilities and resolve multimaster conflicts when Opsware SAS is configured for multimaster mode.

Chapter 3: Opsware Satellite Administration: Provides overview information about an Opsware Satellite facility and how to administer one after installation.

Chapter 4: Opsware SAS Maintenance: Provides information about possible Opsware SAS problems, how to contact support, and how to test and diagnose both Opsware SAS components and managed servers. It describes how to locate component logs, stop and restart Opsware SAS components, and restart order dependencies. It also discusses how to administer the Opsware Access & Authentication Directory.





Conventions in this Guide

This guide uses the following typographical and formatting conventions.

NOTATION	DESCRIPTION
Bold	Identifies field menu names, menu items, button names, and inline terms that begin with a bullet.
<code>Courier</code>	Identifies text that is entered or displayed at the command-line prompt, such as Unix commands, Opsware SAS commands, file names, paths, directories, environment variable names, contents of text files that are viewed or edited with a text editor, source code in a programming language, and SQL (database) commands.
<i>Italics</i>	Identifies document titles, DVD titles, web site addresses. Used to introduce new terms when they are first defined in a document and for emphasis.

Icons in this Guide

This guide uses the following iconographic conventions.

ICON	DESCRIPTION
	This icon represents a note. It identifies especially important concepts that warrant added emphasis.
	This icon represents a requirement. It identifies a task that must be performed before an action under discussion can be performed.
	This icon represents a tip. It identifies information that can help simplify or clarify tasks.
	This icon represents a warning. It is used to identify significant information that must be read before proceeding.

Guides in the Documentation Set and Associated Users

- The *Opsware® SAS User's Guide* is intended to be read by the system administrator who is responsible for performing the day-to-day functions of managing servers, provisioning operating systems, uploading packages, setting up the Software Tree and node hierarchies, attaching software applications and installing them on servers, managing patches, reconciling servers with software, creating and executing scripts, tracking configuration, and deploying and rolling back code and content. It also documents the day-to-day functions of managing servers, such as server compliance and auditing, software packaging, application configuration, agent deployment, and global shell remote data center management.
- The *Opsware® SAS Administration Guide* is intended to be read by Opsware administrators who will be responsible for monitoring and diagnosing the health of the Opsware SAS components.
- The *Opsware® SAS Deployment and Installation Guide* is intended to be used by system administrators who are responsible for the installation of Opsware SAS in a facility. It documents how to run the Opsware Installer and how to configure each of the components.
- The *Planning Deployments for Opsware® SAS* is intended to be used by advanced system administrators who will be responsible for planning all facets of an Opsware SAS installation and deployment. It documents all the main features of Opsware SAS and scopes out the planning tasks necessary to successfully deploy Opsware SAS. Sections include: planning the Opsware SAS design for a core, types of installations, and discusses business goals that can be achieved using the software. It also includes information on system sizing, checklists, and best practices.
- The *Opsware® SAS Configuration Guide* is intended to be used by system administrators who are responsible for all facets of configuring the Opsware Command Center. It documents how to set up users and groups, how to configure Opsware server management, and how to set up the main Opsware Command Center features, such as patch management, configuration tracking, software repository replicator setup, code deployment, and software provisioning.

Opsware, Inc. Contact Information

The main web site and phone number for Opsware, Inc. are as follows:

- <http://www.opsware.com/index.htm>
- +1 (408) 744-7300

For links to the latest product documentation and software downloads, see the Opsware Customer Support site:

- <https://download.opsware.com/opsw/main.htm>

For troubleshooting information, you can search the Opsware Knowledge Base at:

- <https://download.opsware.com/kb/kbindex.jspa>

The Opsware Customer Support email address and phone number follow:

- support@opsware.com
- +1 (877) 677-9273

Chapter 1: Opsware SAS Overview

IN THIS CHAPTER

This section contains the following topics:

- Opsware SAS Technology
- Types of Opsware Users
- Types of Opsware SAS Installations
- Opsware SAS Components
- Interaction Among Opsware SAS Components

Opsware SAS Technology

Opsware SAS provides a core set of features that automate critical areas of server and application operations – including the provisioning, deployment, patching, and change management of servers – across major operating systems and a wide range of software infrastructure and application products.

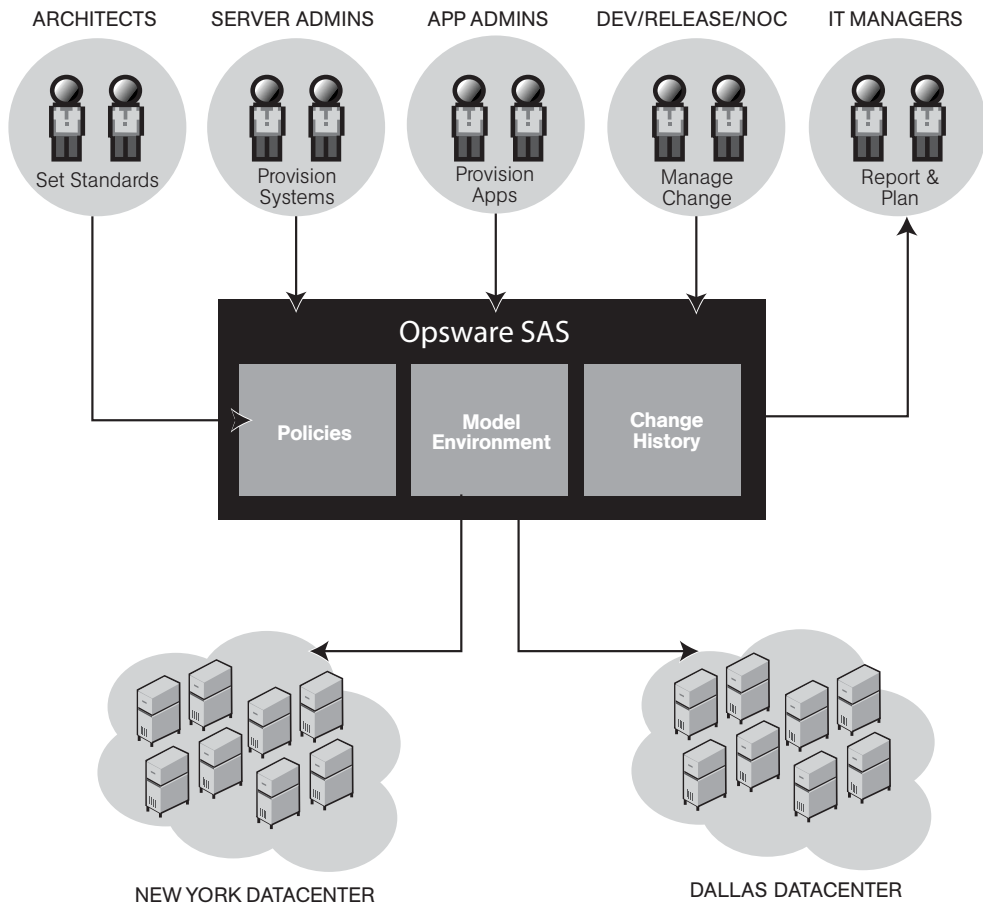
Opsware SAS does not just automate your operations, it also allows you to make changes more safely and consistently, because you can model and validate changes before you actually commit the changes to a server. Opsware SAS helps ensure that modifications to your servers work on your first attempt, thereby reducing the risk of downtime.

Using Opsware SAS, you can coordinate many operations tasks, across many IT groups with everyone working with the same understanding of the state of servers, applications, and configurations. This coordination ensures that all IT administrators have full knowledge of the current state of the environment before further changes are made.

Opsware SAS allows you to incorporate and maintain operational knowledge gained through long hours of trial-and-error processes. After an administrator has found and tested a procedure or configuration, that knowledge can be translated into a model that is stored in a central repository. This allows you to continue to benefit from the operational knowledge gained by your system administrators, even if they are no longer working in your organization.

The following figure provides an overview of how Opsware SAS automates server and application operations across all major platforms and a wide range of applications. Each feature that is shown in the diagram is discussed in the following sections.

Figure 1-1: Overview of Opsware SAS Features



Types of Opware Users

The following table identifies the types of Opware users and their responsibilities.

Table 1-1: Types of Opware Users

OPSWARE USER	RESPONSIBILITIES
Data Center and Operations Personnel	After manually racking and stacking servers, manage customer facilities and boot bare-metal servers over the network or from an Opware boot image.
System Administrators	Install operating systems and applications (for example, Solaris 5.7 or WebLogic 6.0 Web Server), upgrade servers, create operating system definitions, and set up software provisioning.
Site Engineers and Customer Project Managers	Deploy custom code on servers.

In addition to the Opware users listed above, this guide describes the following three types of users:

- **End Users** are responsible for all aspects of managing and provisioning the servers in an operational environment. In the Opware SAS documentation, these users are referred to as Opware users or system administrators. These users log into the Opware Command Center and OCC Client and use these interfaces to manage servers in their IT environment.
- **Opware Administrators** are the users, with special training and information, who are responsible for installing and maintaining Opware SAS. In the Opware SAS documentation, these users are referred to as Opware administrators. They use the Administration features in the Opware Command Center to manage Opware SAS and Opware users (by adding user accounts and assigning permissions for different levels of operation and access), to add customers and facilities, and to change Opware SAS configurations. They monitor and diagnose the health of Opware SAS components. Opware administrators need to understand how Opware SAS features operate to support users and Opware SAS.
- **Policy Setters** are the power users who are responsible for architecting what Opware SAS will do in the managed environment; for example, they determine which operating systems can be installed on your managed servers and how those operating systems

will be configured during installation. Policy setters, for example, prepare specific features in Opware SAS by defining the Software Tree, preparing Operating System Definitions, and acting as Patch Administrators to approve patches for installation in the operational environment.

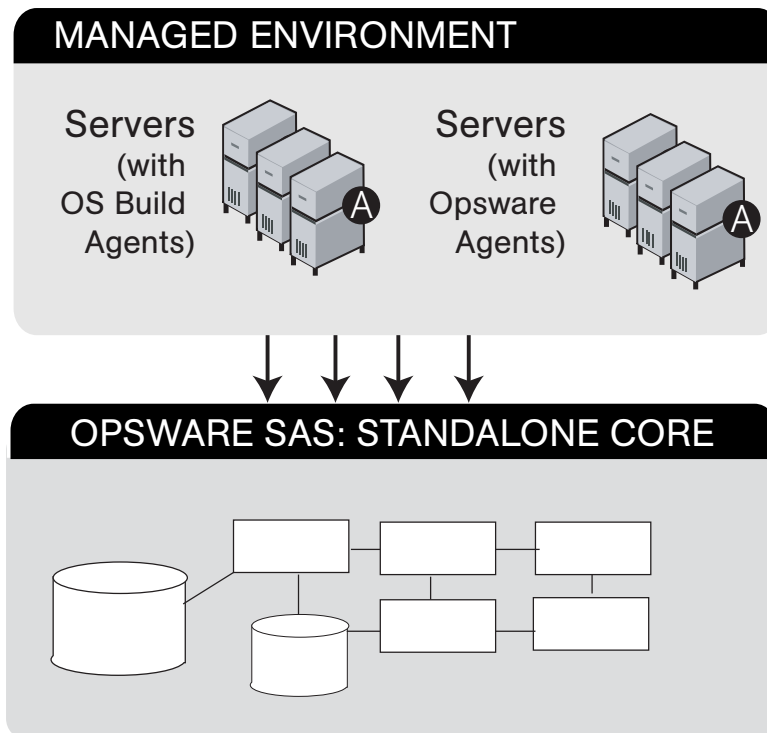
Opware SAS Environment

In an Opware SAS-managed environment, the following two main components are installed in your facility that provide the core Opware SAS platform support and the infrastructure used to run your operational environment:

- **Opware SAS Core Technology:** The set of back-end processes needed to manage the environment such as the Software Repository, the Model Repository, the Command Engine, the Data Access Engine, and so forth.
- **Managed Environment:** All servers that Opware SAS manages by virtue of the Opware Agent, which resides on each managed server and performs tasks such as installing or removing software, installing or removing patches, and so forth. The OS

Build Agent also resides on each server, and is responsible for registering a bare metal server with Opware SAS and guiding the OS installation process. See Figure 1-2.

Figure 1-2: Opware SAS Environment



Model-Based Control

Opware SAS utilizes a model-based control approach to accomplish infrastructure management.

Users and administrators interact with the Opware Command Center, a Web-based front-end application, to accomplish Opware SAS tasks such as server management, software distribution, patch management and installation, inventory reporting, system diagnosis, and code and content deployment to the operational environment. Opware SAS tracks the operational environment through a back-end system and data model that has the following key components:

- **Model Repository:** A data repository that stores information about the hardware and software deployed in the operational environment. All Opware SAS components work from, or update, a data model of information maintained in the Model Repository for all servers that Opware SAS manages.

- **Software Repository:** A central repository for all software that Opware SAS manages and deploys in the operational environment.
- **Command Engine:** A system for running distributed programs across many servers.
- **Opware Agent:** On each Opware SAS-managed server. Whenever Opware SAS needs to enact change on servers or query servers, it sends requests to the Opware Agents.

Types of Opware SAS Installations

There are three basic types of Opware SAS installations: standalone, multimaster, and satellite.

- **Standalone:** A standalone core does not communicate or exchange information with other cores. A standalone core manages servers in a single facility. (Optionally, a standalone core can also manage servers in remote facilities installed with Opware Satellites.) A core contains all components of Opware SAS, except for the Opware Agents, which run on the servers managed by the core.
- **Multimaster:** A multimaster core exchanges information with other cores. This collection of cores is called a multimaster mesh. With a multimaster mesh, you can centralize the management of several facilities but still get the performance benefits of having a local copy of key Opware SAS data at each facility.
- **Satellite:** Installed in a remote facility, an Opware Satellite provides network connection and bandwidth management for a core that manages remote servers. A Satellite must be linked to at least one core, which may be either standalone or multimaster.



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 Opware core or Satellite is associated with a specific facility.

Opsware SAS Components

Opsware SAS has an agent-server architecture. Each server managed by Opsware SAS runs an Opsware Agent, which performs tasks remotely. The server portion of Opsware SAS is called the Opsware core, consisting of multiple, integrated components, each with a unique purpose.

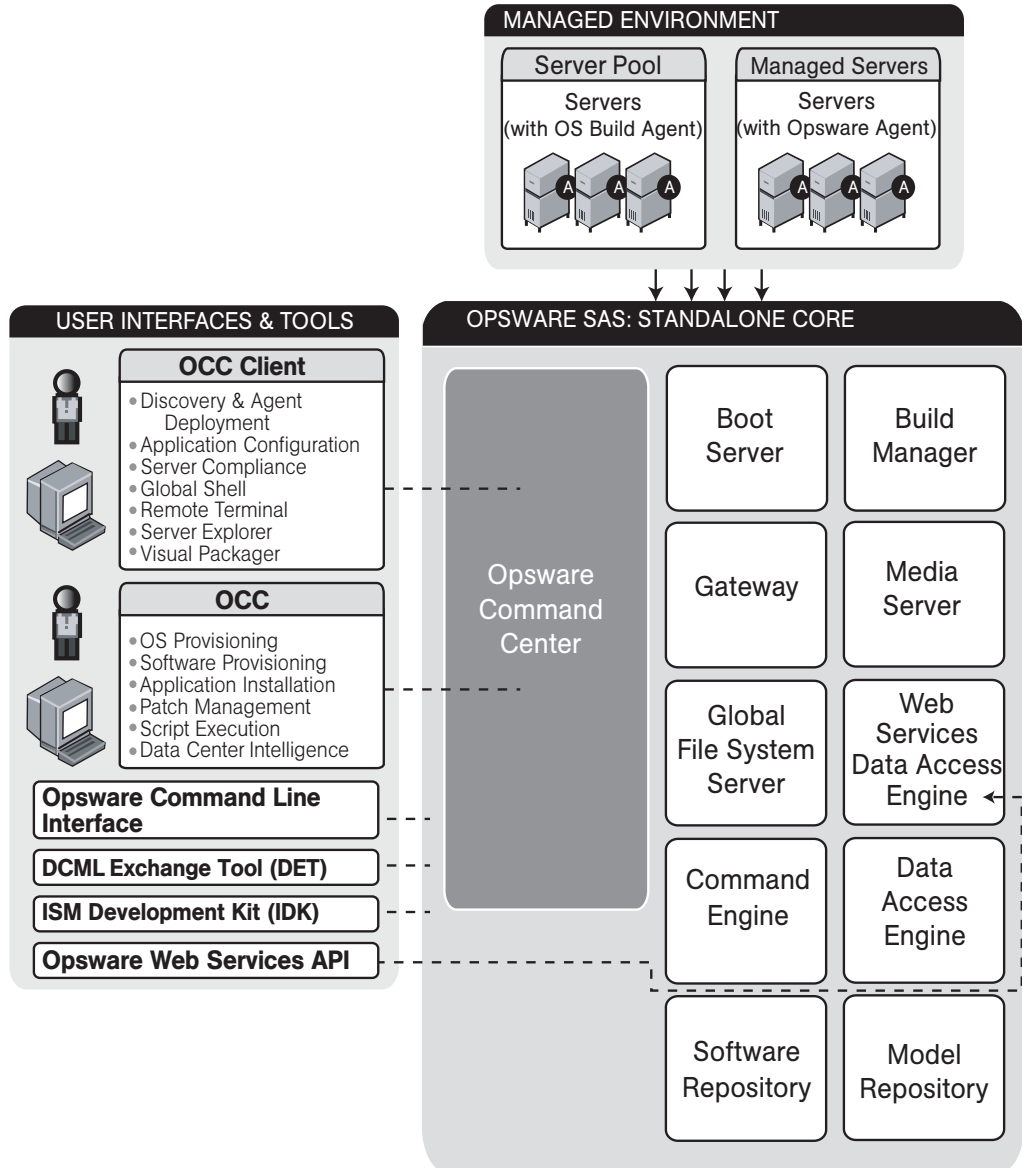
The sections that follow describe the components of Opsware SAS:

- **Boot Server:** Part of the OS Provisioning feature that supports network booting of Sun and x86 systems.
- **Build Manager:** This facilitates communication between components for OS provisioning.
- **Command Engine:** The system for running distributed programs across many servers.
- **Data Access Engine:** The XML-RPC interface to the Model Repository.
- **Media Server:** This server provides network access to vendor-supplied media used during OS provisioning.
- **Model Repository:** The Opsware SAS data repository (database).
- **Model Repository Multimaster Component:** The application that propagates and synchronizes changes from each Model Repository database to all other Model Repository databases.
- **Opsware Agents:** Intelligent agents that run on each server that Opsware SAS manages.
- **Opsware Command Center:** The user interface to Opsware SAS.
- **OS Build Agent:** The agent responsible for registering a bare metal server with Opsware SAS and guiding the OS installation process.
- **Software Repository:** The central repository for all software that Opsware SAS manages.
- **Software Repository Replicator:** This serves as backup for Software Repositories in a multimaster mesh, ensuring that packages are available, even if one of the Software Repositories becomes unavailable.
- **Software Repository Multimaster Component:** This aids in transferring software from the Software Repository in one facility to the Software Repository in another facility in a multimaster mesh.

- **Software Repository Cache:** This contains local copies in the Opware Satellite of the Software Repository of the core (or another Satellite).
- **Web Services Data Access Engine:** This provides increased performance from the Model Repository to other Opware SAS components.
- **Opware Gateway:** This provides network connectivity to Opware cores and Satellites.
- **Global File System Server:** This dynamically constructs the Opware Global File System (OGFS), a virtual file system.

The following figure shows an overview of Opware SAS components in a standalone core. The components in a core can be distributed across multiple servers.

Figure 1-3: Overview of the Opware Components



Boot Server

The Boot Server, part of the OS Provisioning feature, 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, and Sun Solaris TFTP and NFS.

Build Manager

The Build Manager component facilitates communications between OS Build Agents and the Command Engine. It accepts OS provisioning commands from the Command Engine, and it provides a runtime environment for the platform-specific build scripts to perform the OS provisioning procedures.

Command Engine

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

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

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 Opware Command Center, system data collection, and monitoring agents on servers.

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 Opware SAS without requiring system-wide changes.

Media Server

The Media Server is also part of the OS Provisioning feature, and 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 NFS.

Model Repository

The Model Repository is implemented as an Oracle database. All Opsware SAS components work from, or update, a data model maintained for all servers that Opsware SAS manages. The Model Repository contains essential information necessary to build, operate, and maintain the following items:

- A list of all servers under management.
- The hardware associated with these servers, including memory, CPUs, storage capacity, and so forth.
- The configuration of those servers, including IP addresses.
- The operating system, system software, and applications installed on servers.
- Information on other software available for installation on servers and how it is bundled
- Authentication and security information.

Each Opsware core, whether standalone or multimaster, contains a single Model Repository. An Opsware Satellite, which relies on a core, does not contain a Model Repository.

Model Repository Multimaster Component

The Model Repository Multimaster Component is installed in a core that belongs to a multimaster mesh. The Model Repository Multimaster Component synchronizes the data in the Model Repositories of the mesh, propagating changes from one repository to another. Every Model Repository instance has one Model Repository Multimaster Component instance. The Model Repository Multimaster Component uses TIBCO Rendezvous.

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. The receiver (Inbound Model Repository Multimaster Component) accepts the transactions and applies them to the local Model Repository.

Opsware Agents

Each server that Opsware SAS manages has an intelligent agent running on that server. The Opsware Agent is the agent of change on a server. Whenever Opsware SAS needs to make changes to servers, it does so by sending requests to the Opsware Agent.

Depending on the request, the Opsware Agent might use global Opsware SAS services (such as the Model Repository and Software Repository) in order to fulfill the request.

Some functions that the Opsware Agent supports are:

- Software installation and removal
- Configuration of software and hardware
- Periodically reporting server status
- Auditing of the server

An Opsware Agent is idle unless Opsware SAS is trying to perform some change on the server. In addition, each Opsware Agent periodically contacts the Model Repository and registers itself, which allows the Model Repository to keep track of machine status, and know when particular servers are disconnected from and reconnected to the network.

Dormant Opsware Agents

The Opsware Agent Installer can install Opsware Agents even when Opsware SAS core is not available to a server. If a newly-installed Opsware Agent cannot contact an Opsware SAS core, the Opsware Agent runs in a dormant mode. While dormant, it periodically attempts to contact Opsware SAS core.

When Opsware SAS core becomes available, the Opsware Agent performs the initialization tasks, such as hardware and software registration, that usually take place when the Opsware Agent is first installed.

Opsware Command Center

The Opsware Command Center is a user interface to Opsware SAS. Through the web-based user interface, an Opsware SAS user can provision and maintain systems, and deploy code and content to servers. An Opsware administrator adds users and defines access to specific Opsware SAS resources.

The Opsware Command Center talks primarily to the Data Access Engines (which communicate with the Model Repository), though they also talk directly to other back-end services to implement some operations. Users accessing the Opsware Command Center are authenticated before they gain access.

OS Build Agent

The OS Build Agent, part of the OS Provisioning feature, is responsible for registering bare metal servers in Opware SAS. In addition, it is the agent of change on the server during the OS installation process (that the Build Manager manages) until the actual Opware Agent is installed.

Software Repository

The Software Repository is the central repository for all software that Opware SAS manages. It contains packages for operating systems, applications (for example, BEA WebLogic or IBM WebSphere), databases, customer code, and software configuration information.

Working with the Software Repository, an Opware Agent can install software running on the server where the Opware Agent is installed. The Model Repository then updates its record of the software installed on the server. This process of updating the actual software configuration of a server with a specified configuration stored in the Model Repository is called reconciliation.

You can install new software, code, or configurations in the Software Repository by first packaging the files, and then uploading them into the Software Repository.

See the *Opware[®] SAS Configuration Guide* for information about how to upload software packages to the Software Repository.

Software Repository Replicator

The Software Repository Replicator provides backup functionality for Software Repositories running in a multimaster mesh. In most deployments, the Software Repositories do not all have the same content. If one of the Software Repositories becomes unavailable, this might result in some packages not being available until the Software Repository is back online.

Using the Software Repository Replicator provides redundant storage of Software Repositories and thereby helps to ensure that all packages remain available even when a Software Repository goes offline.

Software Repository Cache

Installed in an Opware Satellite, a Software Repository Cache contains local copies of the contents of the Software Repository of the core (or of another Satellite). These local copies improve performance and decrease network traffic when the core installs or updates software on the managed servers in the Satellite.

Software Repository Multimaster Component

The Software Repository Multimaster Component allows software to be distributed across several Software Repositories and to be transferred from one repository to another on-demand. For example, a Solaris package that resides on Software Repository (A) is needed for installation in a second facility that contains Software Repository (B), which is part of the same multimaster mesh. The Multimaster Component allows B to discover the presence of the package on A. The package is then transferred and cached at B so that it can be used in the second facility.

Web Services Data Access Engine

The Web Services Data Access Engine provides a public object abstraction layer to the Model Repository. This object abstraction can be accessed through a Simple Object Access Protocol (SOAP) API by third-party integration components, or it can be accessed through a binary protocol by Opware SAS components like the Opware Command Center. It provides increased performance to other Opware SAS components.

Opware Gateway

The Opware Gateway allows an Opware core to manage servers that are behind one or more NAT devices or firewalls. Connectivity between gateways is maintained by routing messages over persistent TCP tunnels between the gateway instances.

Additionally, the gateway provides network bandwidth management between Opware cores in a multimaster mesh and between cores and Satellites. The ability to manage network bandwidth is important when a tunnel between gateway instances transits a low-bandwidth link, which might be shared with a bandwidth-sensitive application.

One or more Opware Gateways service the managed servers contained within an Opware realm. In Opware SAS, a realm is a routable IP address space, which is serviced by one or more gateways. All managed servers that connect to an Opware core via a gateway are identified as being in that gateway's realm.

Global File System Server

The Opsware Global Shell feature runs on the Global File System Server, which dynamically constructs a virtual file system – the Opsware Global File System (OGFS). The Global File System Server component is installed on a Linux server in an Opsware core. The Global File System Server can connect to an Opsware Agent to open a Unix shell or a Windows Remote Desktop connection on a managed server.

Interaction Among Opsware SAS Components

To understand Opsware SAS architecture, review the following types of Opsware SAS component interactions:

- General Interaction Among Components
- Opsware SAS Security
- OS Provisioning
- Patch Management
- Software Provisioning
- Code Deployment and Rollback
- Script Execution
- Integration with AIX and HP-UX Installation Technology
- Component Interaction in Multiple Facilities
- Discovery and Agent Deployment
- Application Configuration Management
- Visual Packager
- Server Compliance and Auditing

General Interaction Among Components

The Opsware Command Center, Command Engine, Software Repository, and Opsware Agent interact with the Model Repository through the Data Access Engine.

The Data Access Engine issues queries against the Model Repository. It does not cache query results.

The Software Repository authenticates all clients. It maps the client's IP address to the customer name. The Software Repository performs this mapping to enforce access rules on customer-specific files.

Opware SAS Security

To enable secure communication with the Opware Agent, Opware SAS automatically issues a unique cryptographic certificate to every server that it manages. The certificate is tied to the server to which it is issued, and cannot be copied and used by a different server. The certificate allows the Opware Agent to establish a secure https connection to Opware SAS components.

As an additional security measure, Opware SAS performs checks on all requests that an Opware Agent issues. Opware SAS verifies that the requested operation is appropriate for the particular server and checks the parameters of the request to make sure that they fall within reasonable bounds.

OS Provisioning

The OS Provisioning feature supports installation-based provisioning using Red Hat Linux Kickstart, Sun Solaris JumpStart, and Microsoft Windows unattended installation. Image-based provisioning (using Symantec Ghost and Sun Solaris Flash) is not supported out-of-the-box.

Because the OS Provisioning feature supports installation-based provisioning, your organization can keep its OS installations lean. Rather than trying to manage changing software through master images, you can use the OS Provisioning feature to install and remove often changing software, including system patches, system utilities, and third-party agents (such as monitoring, backup, and anti-viral agents). See the *Opware® SAS User's Guide* for information about the OS provisioning process.

Figure 1-4: OS Provisioning Step 1: Initial Booting

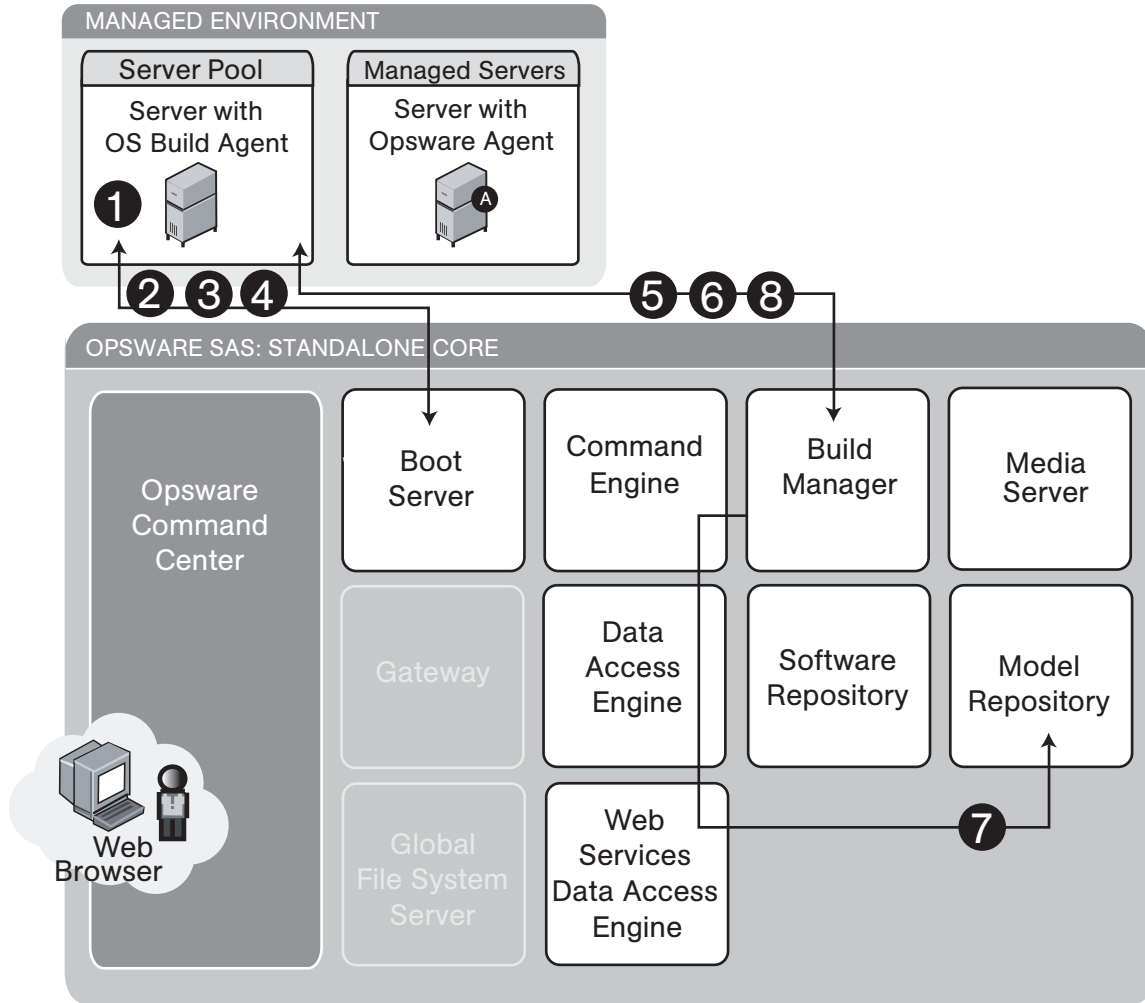


Figure 1-5: OS Provisioning Step 2: OS Installation with Wizard

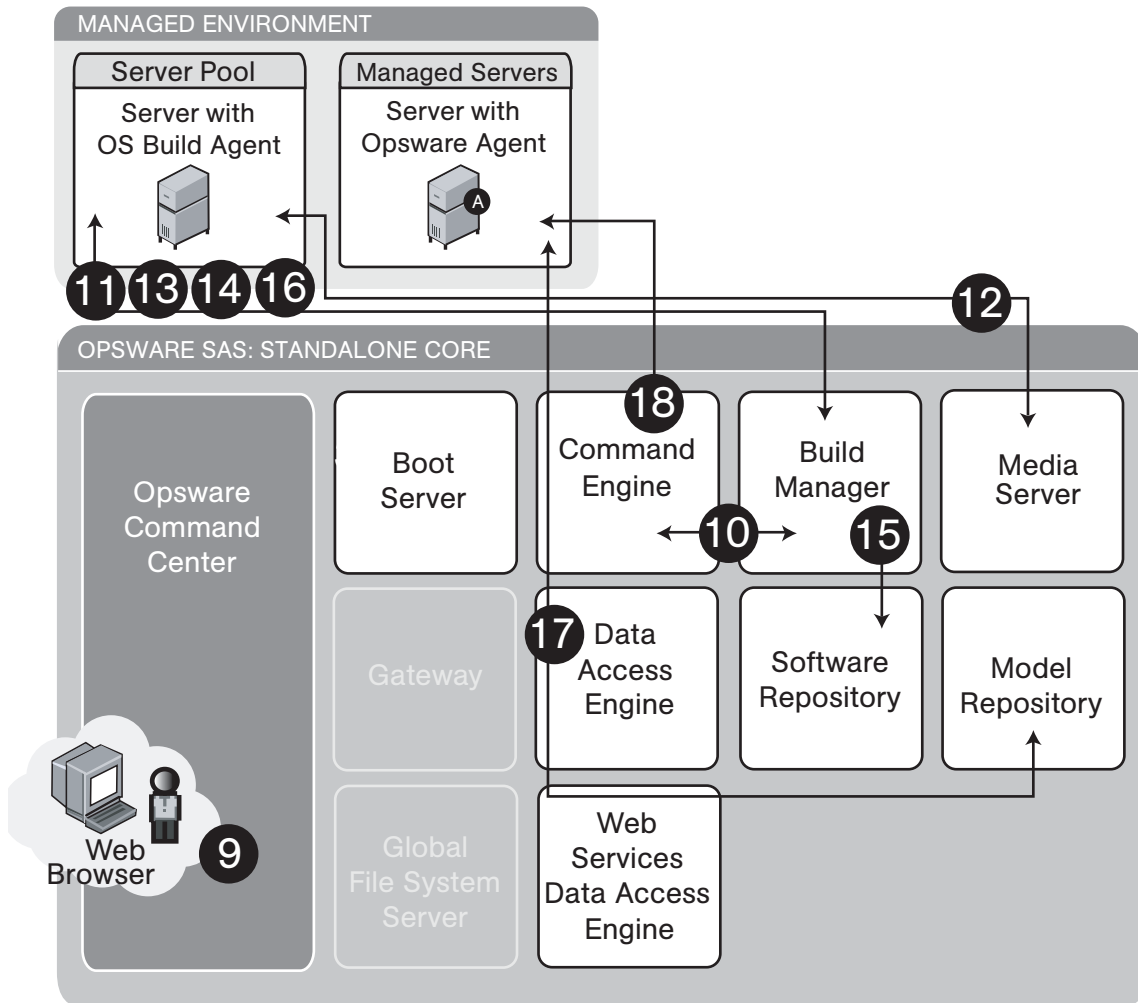


Figure 1-4 and Figure 1-5 illustrate the OS provisioning process:

OS Provisioning Step 1: Initial Booting:

- 1** The DHCP request is a network broadcast.
- 2** The DHCP reply contains the IP address of the Build Manager for use by Sun Solaris and Red Hat Linux provisioning. For Microsoft Windows provisioning, the DNS configuration in the DHCP reply must resolve the host name `buildmgr`.

- 3** TFTP is used to boot the server over the network (using inetboot for Solaris, and PXE for Windows and Linux). Instead of PXE, Windows can use a boot floppy and Linux can use a boot CD.
- 4** An NFS boot image is used by Solaris and Linux only.
- 5** The OS Build Agent pings the Build Manager.
- 6** The Build Manager invokes a Build Script that probes the server's hardware.
- 7** The server is registered with Opsware SAS.
- 8** The OS Build Agent periodically contacts the Build Manager with a ping message. The system remains in this state until a user provisions an OS onto the server with the Opsware Command Center or until the server is removed from the network.

OS Provisioning Step 2: OS Installation with Wizard:

- 9** A user initiates OS provisioning with the Install OS Wizard in the Opsware Command Center.
- 10** Feedback is provided throughout OS provisioning with status messages passed from the Build Manager to the Command Engine and from the Command Engine to the Opsware Command Center.
- 11** A Media Resource Locator contains the network location (host name and path) of an NFS or SMB server from which to retrieve the vendor OS installation media.
- 12** The installation media is mounted with NFS (Solaris and Linux) or SMB (Windows).
- 13** The vendor installation program is used to install the OS (Sun Solaris Jumpstart, Red Hat Linux Kickstart, or Windows unattended.txt).
- 14** The server is rebooted after OS installation.
- 15** The OS Build Agent gets a copy of the Opsware Agent from the Software Repository.
- 16** The OS Build Agent is used to install the Opsware Agent.
- 17** Hardware and software registration is performed as part of the Opsware Agent installation.
- 18** The Reconcile function installs additional software that the vendor installation program did not install.

Steps 11 through 17 are managed by a build script that runs inside the Build Manager. The build script is invoked by the provisionOS script and manages the OS installation at a micro level. The provisionOS script is run by the Command Engine and is responsible for managing the installation process at a macro level.

Patch Management

Opware SAS automates the key aspects of patch management, while offering a fine degree of control over how and under what conditions patches are installed.

Because patches are often released to address grave security threats, an organization needs to be able to roll out patches quickly, before systems become compromised. At the same time, however, patches can cause serious problems, from performance degradation to general system failure.

The Patch Management feature allows you to react quickly to newly discovered threats, but it also provides support for strict testing and standardization of patch installation. And, if patches later cause problems even after being tested and approved, the Patch Management feature also allows you to uninstall the patches in a safe and standardized way. See Figure 1-6 through Figure 1-9.

See the *Opware® SAS User's Guide* for information about the patch management process.

Figure 1-6: Patch Management Feature: Upload Unix Patch

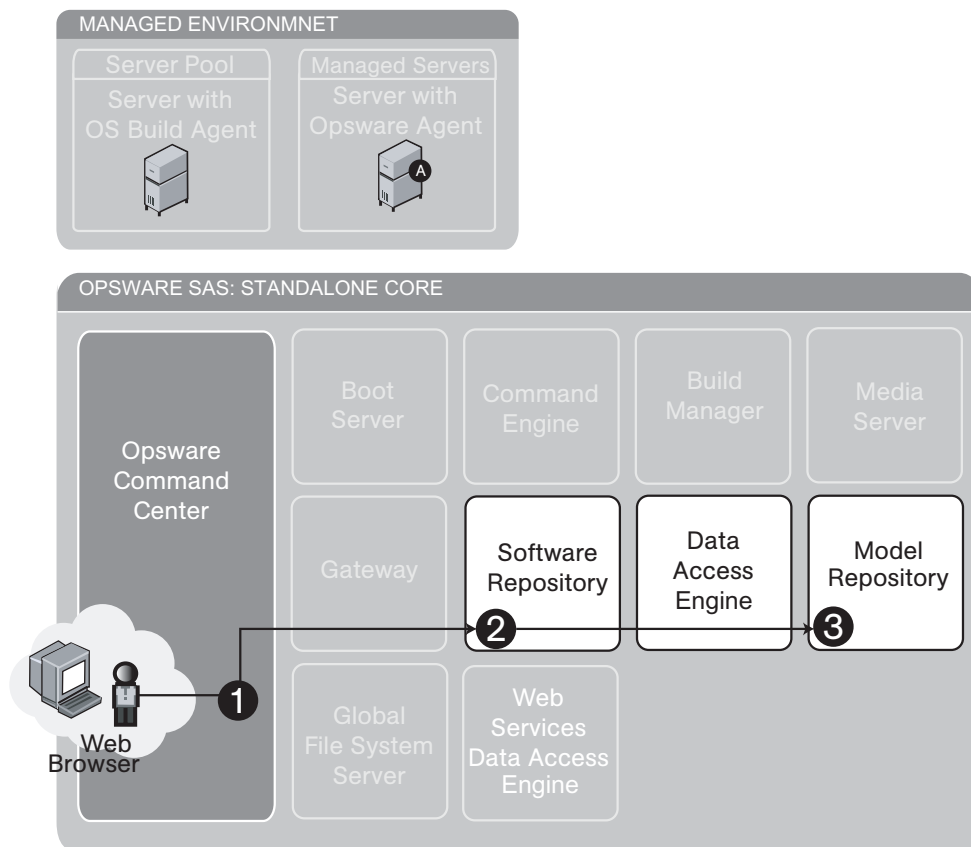


Figure 1-6 illustrates the following upload patch process for Unix patches:

- 1** An Opware user with the required permissions logs into the Opware Command Center and clicks the Upload Patch wizard link on the Home page.
- 2** Using the wizard, the user specifies a Patch type and Platform and uploads the file to the Software Repository.

- 3 The Software Repository places a record of the location, file size, and patch state of each patch in the Model Repository via the Data Access Engine.

Figure 1-7: Patch Management Feature: Install Unix Patch

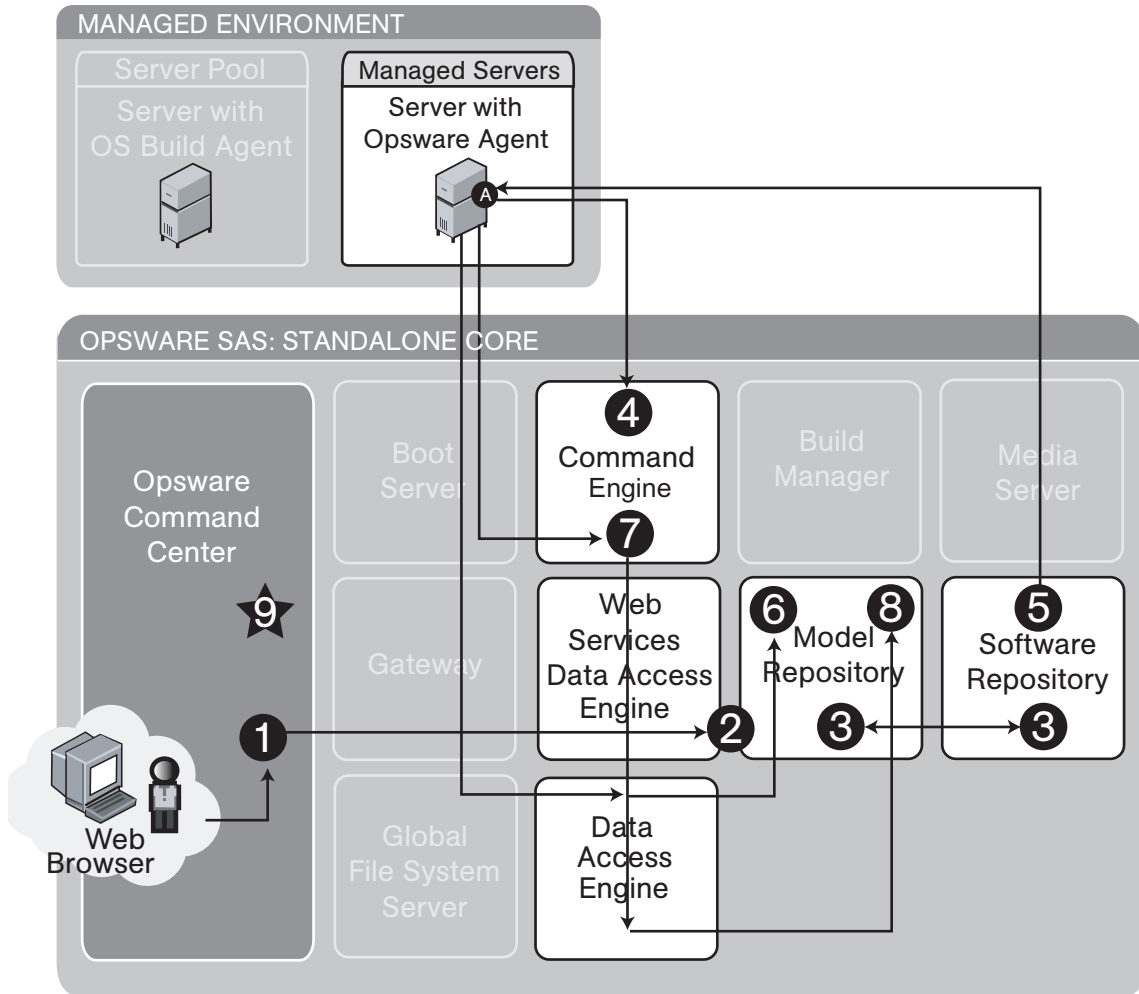


Figure 1-7 illustrates the install patch process for Unix patch:

- 1 An Opware user with the required permissions logs into the Opware Command Center and clicks the Install Patch wizard link on the Home page.
- 2 Using the wizard, the user specifies patches and servers and starts the install process, retrieving patch information from the Model Repository via the Web Services Data Access Engine.

- 3** The Software Repository places a record of the location, file size, and patch state of each patch in the Model Repository via the Data Access Engine.
- 4** The Command Engine gets a list of installed software from the Opware Agent on the managed servers. It compares it to the user-specified list of patches to determine what needs to be installed.
- 5** The Opware Agent on each managed server downloads patches from the Software Repository and installs them, performing all required install operations and reboots.
- 6** When installation is complete, a record of all currently-installed software is stored in the Model Repository via the Data Access Engine.
- 7** The Opware Agent on each managed server reports installation status to the Command Engine.
- 8** The Command Engine stores installation status in the Model Repository via the Data Access Engine.
- 9** Operation complete status displays in the Opware Command Center.

Figure 1-8: Patch Management Feature: Uninstall Unix Patch

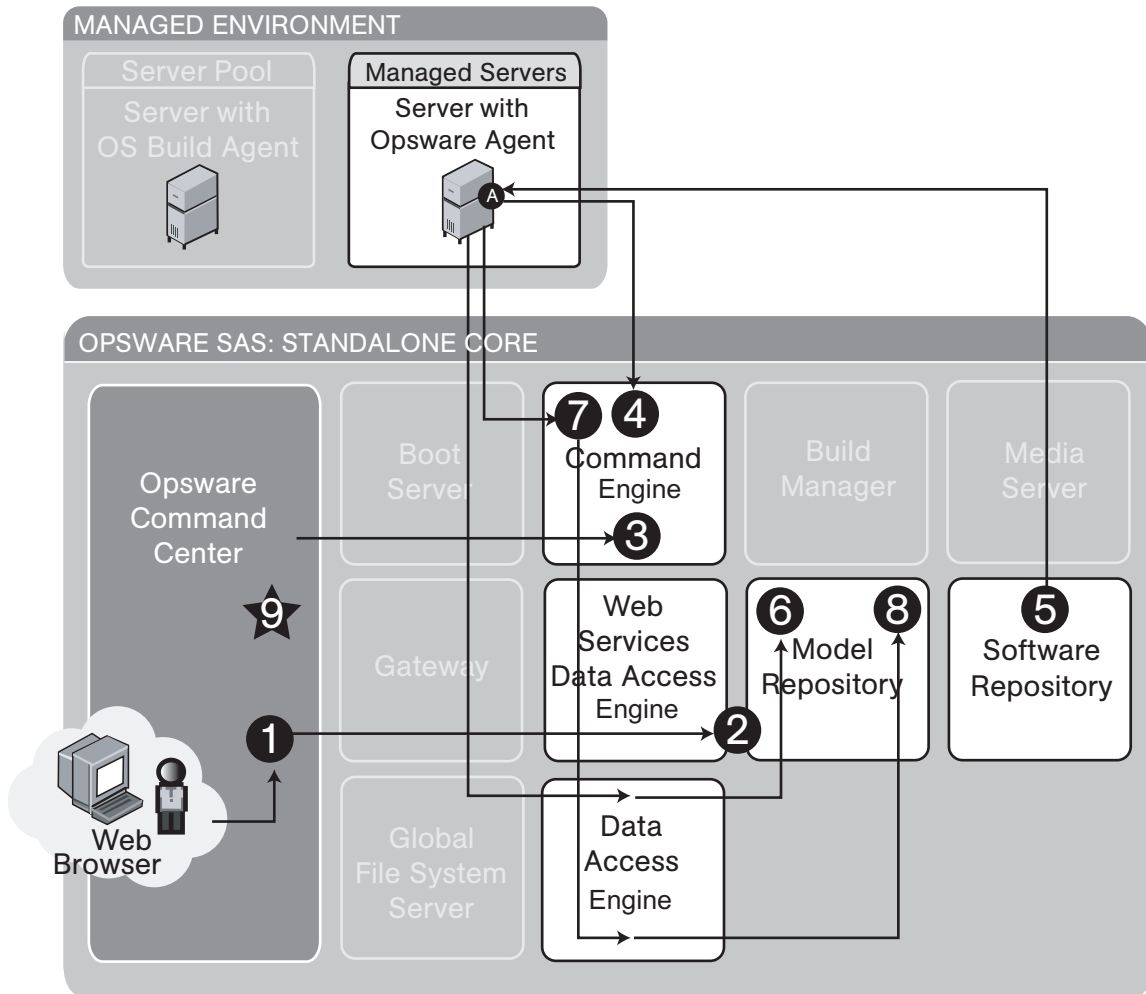


Figure 1-8 illustrates the uninstall patch process for Unix patches:

- 1** An Opware user with the required permissions logs into the Opware Command Center and clicks the Uninstall Patch wizard link on the Home page.
- 2** Using the wizard, the user specifies servers and a patch to be uninstalled and starts the uninstall process, retrieving server and patch information from the Model Repository via the Web Services Data Access Engine.
- 3** The Opware Command Center passes uninstall operation details to the Command Engine.

- 4** The Command Engine gets a list of installed software from the Opware Agent on the managed servers. It compares it to the user-specified patch to be uninstalled and determines if it does need to be uninstalled.
- 5** The Opware Agent on each managed server removes the patch from the managed servers and performs all required uninstall operations and reboots.
- 6** When uninstallation is complete, a record of all currently-installed software is stored in the Model Repository via the Data Access Engine.
- 7** The Opware Agent on each managed server reports uninstallation status to the Command Engine.
- 8** The Command Engine stores uninstallation status in the Model Repository via the Data Access Engine.
- 9** Operation complete status displays in the Opware Command Center.

Figure 1-9: Patch Management Feature: Microsoft Patch Update

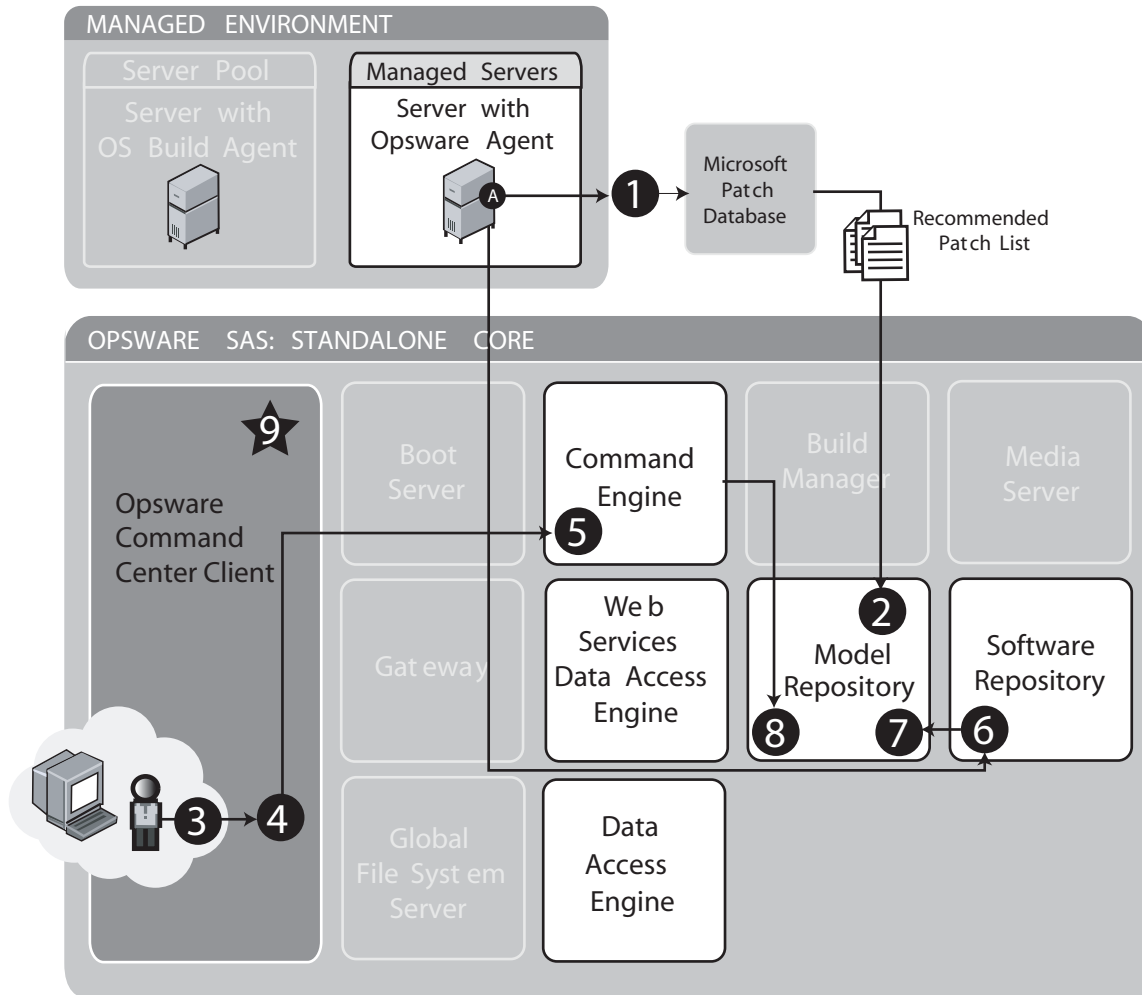


Figure 1-9 illustrates the Microsoft patch update process:

- 1 Every 24 hours, the Opware Agent builds an inventory of software installed on the server. It uses that inventory and the Microsoft Patch Database to determine what Hot fixes and Service Packs are needed to bring the server up to current patch level. This is the Recommended Patch List.
- 2 The Recommended Patch List and a full inventory of installed software is stored in the Model Repository via the Data Access Engine.
- 3 An Opware user with the required permissions logs into the Opware Command Center Client and attaches the Vendor-recommended patch policy to the server.

- 4 Using the Reconcile Patches wizard, the user performs the reconcile policy process to install the patches in the Vendor-recommended patch policy.
- 5 The installation details are passed from the OCC Client to the Command Engine, which obtains a list of installed software from the Opware Agent. It compares this list to the user-selected list and determines what actually needs to be installed.
- 6 The Opware Agent on the managed server downloads patches from the Software Repository and installs them, performing all required install operations and reboots.
- 7 When installation is complete, a record of all currently installed software is stored in the Model Repository via the Data Access Engine.
- 8 Install operation status is reported to the Command Engine, which places it in the Model Repository via the Data Access Engine.
- 9 Operation complete status displays in the Opware Command Center.

Software Provisioning

In Opware SAS, packages reside in a central Software Repository. Opware administrators upload the packages and also specify options that help ensure that the software is installed in a safe and consistent way. Administrators can add pre- and post-install and uninstall scripts to a software package that help control the way that the software is installed. The administrator can use Software Tree nodes to specify dependencies for package installation order. If a user tries to install software on a server that does not have the required prerequisite software, the user is alerted about the need to install the software and the installation operation is stopped on that server. (The installation process continues, however, if other selected servers have the required software already installed on them.)

Opware SAS maintains detailed information about the state of every server under management in a central database called the Model Repository. This information includes details about software that is installed. You can use the information to check the rollout of software and also to help diagnose common server problems.

Information about the software is consolidated into the centralized Model Repository. Provisioning a server with software entails using automation to deliver software and configuration information about applications to any Opware SAS-enabled server.

See Figure 1-10 and Figure 1-11. See the *Opware® SAS User's Guide* for information about the software provisioning process.

Figure 1-10: Software Provisioning Step 1: Preview Reconcile

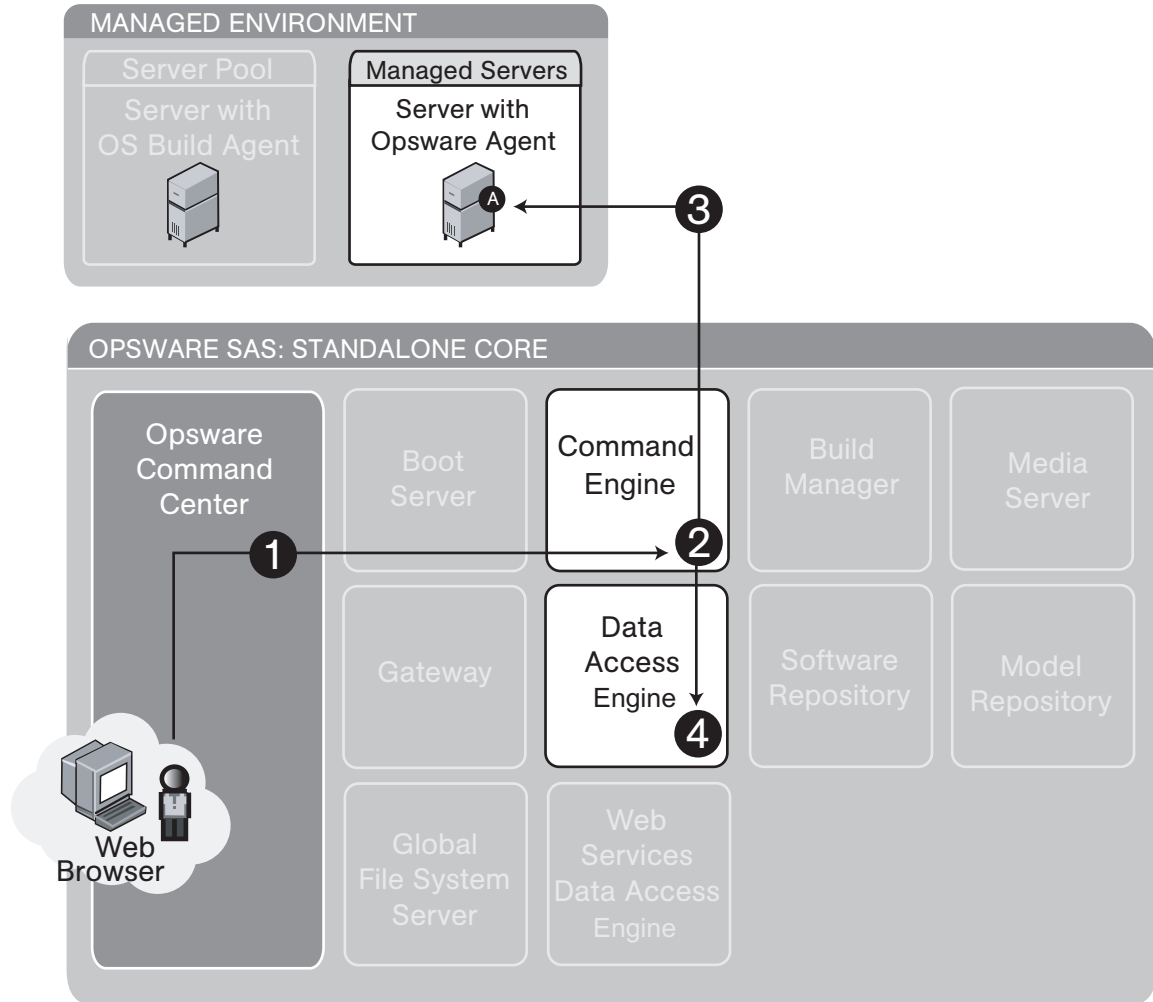


Figure 1-11: Software Provisioning Step 2: Software Installation and/or Removal Through Reconcile

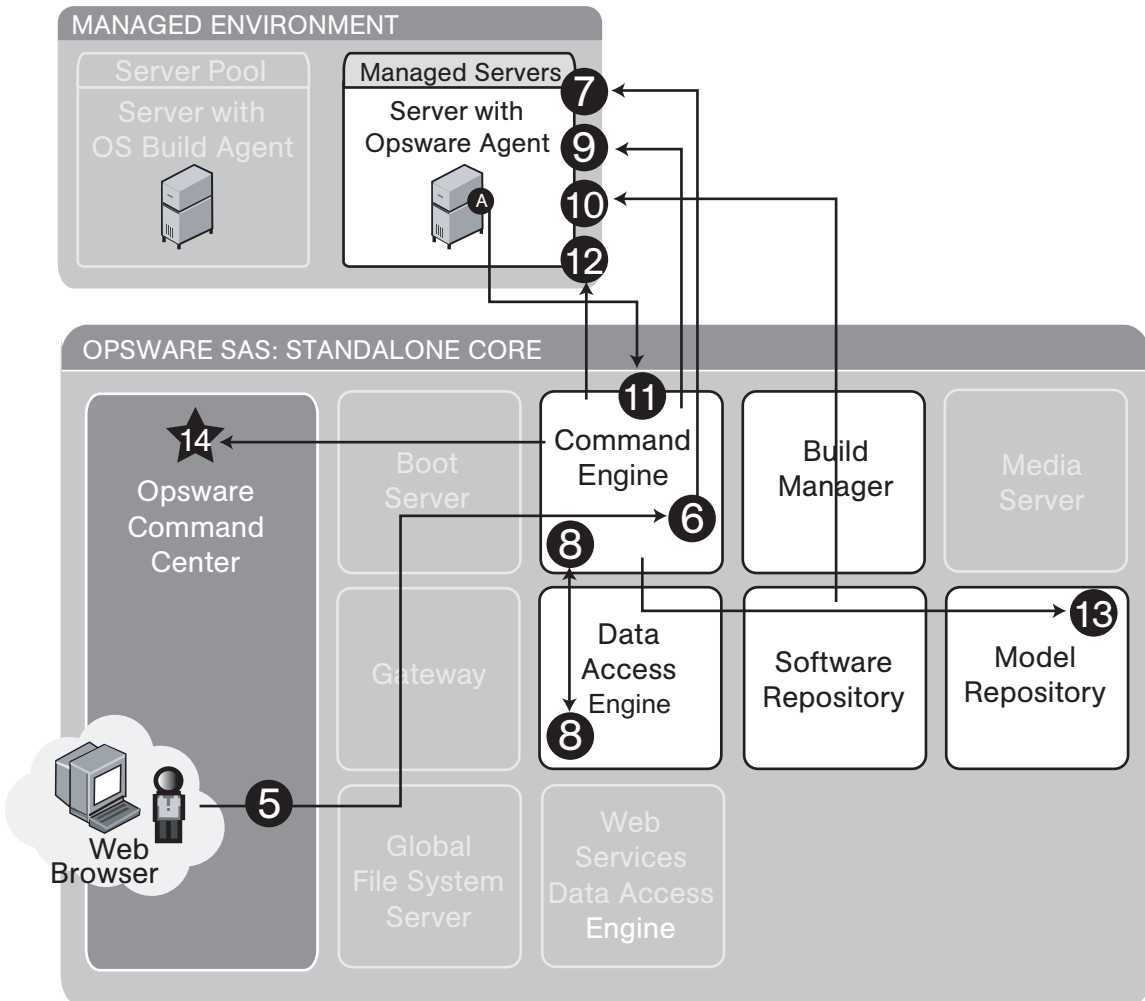


Figure 1-10 and Figure 1-11 illustrate the software provisioning process:

Software Provisioning Step 1: Determine Server Configuration:

- 1** An Opsware user logs into the Opsware Command Center and selects one or more servers and software nodes to reconcile against.
- 2** The Opsware Command Center starts the Preview Reconcile Job in the Command Engine.
- 3** The Opsware Agent on each of the specified servers is queried by the Command Engine for a list of packages installed on its server.

- 4 The Command Engine, via the Data Access Engine, compares that list to the user-specified list of software nodes to determine what needs to be installed or removed.

Software Provisioning Step 2: Software Installation through Reconcile

- 5 At the end of Reconcile Preview, the OCC displays a list of the software to be installed and/or removed. The user confirms proceeding with the reconcile.
- 6 The OCC starts the Reconcile job in the Command Engine.
- 7 The Opsware Agent on each of the specified servers is queried by the Command Engine for a list of packages installed on its server.
- 8 The Command Engine, via the Data Access Engine, compares that list to the user-specified list of software nodes to determine what needs to be installed or removed.
- 9 The Command Engine tells the Opsware Agent to install and/or remove packages.
- 10 The Opsware Agent downloads packages from the Software Repository, removes any packages that need to be removed, and installs the new packages, performing all necessary install, uninstall, and reboots, if required.
- 11 The Opsware Agent reports installation status to the Command Engine.
- 12 The Opsware Agent on each of the specified servers is queried by the Command Engine for a list of packages installed on its server to confirm what was installed and/or removed.
- 13 The Command Engine stores installation status in the Model Repository via the Data Access Engine.
- 14 Status of completed installation and removal of packages displays in the OCC via the Command Engine.

Code Deployment and Rollback

Before you use Code Deployment and Rollback (CDR) to push code and content, you must upload new or updated files to your Opsware SAS staging environment. You can use Opsware SAS-supported content management tools, such as OpenDeploy, scp, or rsync over SSH, to do that.

After you upload the files and test your changes, you can synchronize updates to the production hosts that run your operational environment. You can run specific synchronizations and perform other service deployment operations by selecting CDR menu options available from the Opsware Command Center navigation panel. Figure 1-12 shows the code deployment and rollback process.

See the *Opware® SAS User's Guide* for information about the process to deploy code and content to servers in the managed environment.

Figure 1-12: Code Deployment and Rollback Feature

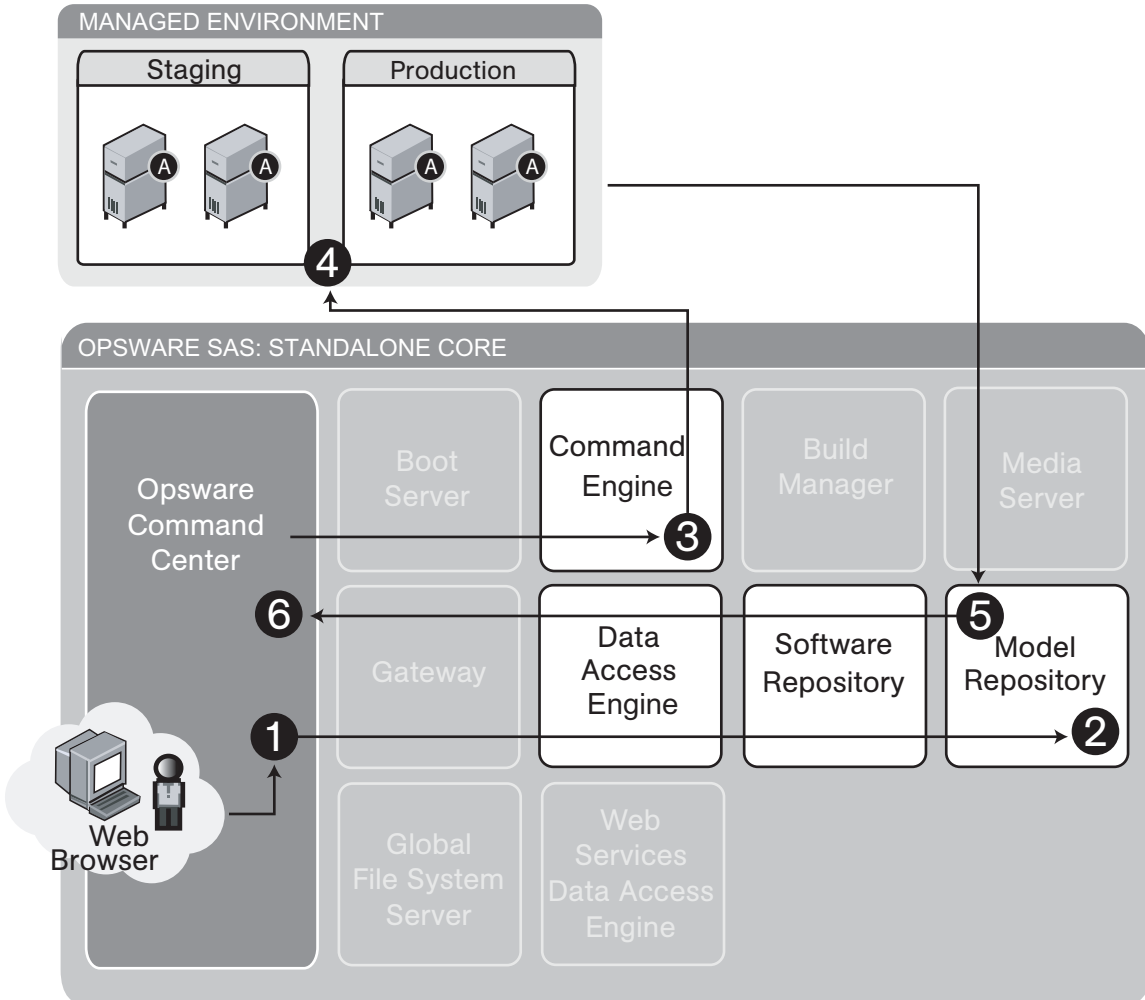


Figure 1-12 illustrates the code deployment and rollback process:

- 1** An Opware user with the required permissions logs into the Opware Command Center, clicks the Deploy Code link, selects a code deployment action, and clicks **Run**.
- 2** The Opware Command Center gets code deployment details from the Model Repository via the Data Access Engine.

- 3** The Opsware Command Center sends code deployment details to the Command Engine.
- 4** The Command Engine sends commands to staging and production servers.
- 5** Results of the code push are sent back to the Model Repository via the Data Access Engine.
- 6** The user views results of the code push.

Script Execution

The Script Execution feature provides features and tools for automating the management and execution of server scripts. Previously, a user created a script and then manually executed the script at individual servers, one server after another. With the Script Execution feature, a user performs all script tasks at one location – the Opsware Command Center.

From the Opsware Command Center, you can create or upload a script, set it up to run simultaneously across multiple Unix or Windows servers, and monitor it as it executes on each server. After a script runs, job- and server-specific execution results are available for review. You can modify, delete, or rerun a script at a later date. See Figure 1-13 and Figure 1-14.

See the *Opware® SAS User's Guide* for information about the process to create and execute scripts in the managed environment.

Figure 1-13: Scripting Feature: Upload Script

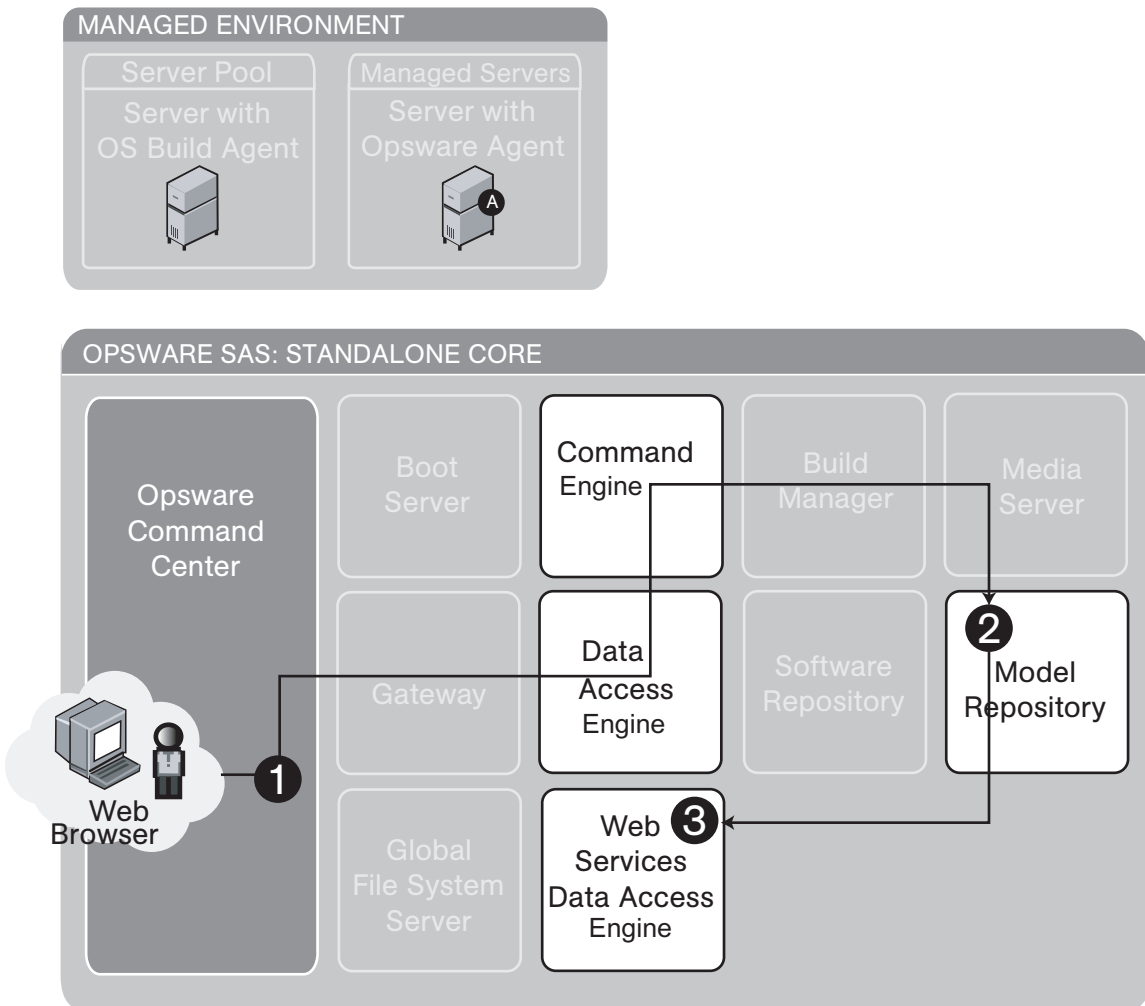


Figure 1-13 illustrates the script execution – upload script process:

- 1** An Opware user with the required permissions logs into the Opware Command Center and clicks the Scripts link under Software and then clicks **New Script**.
- 2** The user clicks **Upload Script**, defines the path, enters Usage Notes, and clicks **Save**. The script is uploaded and saved in the Model Repository by the Command Engine via the Data Access Engine.

- 3 The Web Services Data Access Engine displays the newly uploaded script in the list of available scripts.

Figure 1-14: Scripting Feature: Execute Script

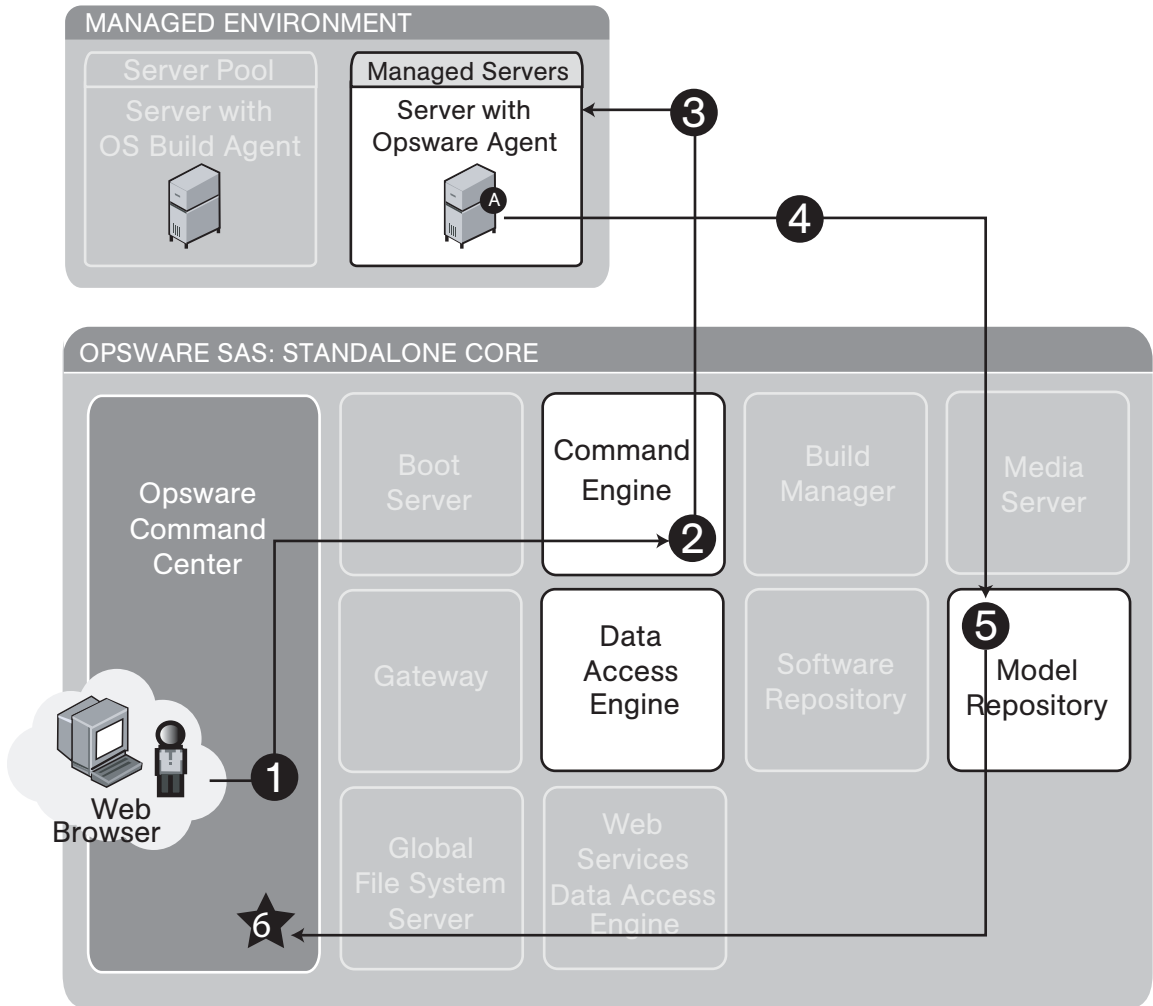


Figure 1-14 illustrates the scrip execution – execute script process:

- 1 An Opware user with the required permissions logs into the Opware Command Center and clicks the Run Distributed Script Wizard link on the home page.
- 2 The user selects the scripts and the servers on which to execute the script and clicks **Run Script**. The request is passed to the Command Engine.

- 3** The Command Engine contacts the Opware Agent on the selected servers and tells it to execute the script.
- 4** The Opware Agent runs the script and sends the results back to the Command Engine.
- 5** The Command Engine aggregates the scripts and stores them in the Model Repository via the Data Access Engine.
- 6** The Model Repository sends the results to the Opware Command Center via the Data Access Engine for the user to view.

Integration with AIX and HP-UX Installation Technology

Integrating Opware SAS with an OS installation technology enables installing an OS by using vendor utilities and automatically installing the Opware Agent, which registers servers' initial configurations with the Model Repository.

See the *Opware[®] SAS Configuration Guide* for information about how Opware SAS integrates with third-party OS installation technologies.

Figure 1-15 explains the interaction between Opware SAS components when Opware SAS is integrated with AIX NIM and HP-UX Ignite OS installation technologies. Opware SAS installation integration with AIX NIM and HP-UX Ignite occurs with the integration of the Opware Installer. See the *Opware® SAS Configuration Guide* for more information.

Figure 1-15: Opware Integration with AIX and HP-UX OS Installation Technology

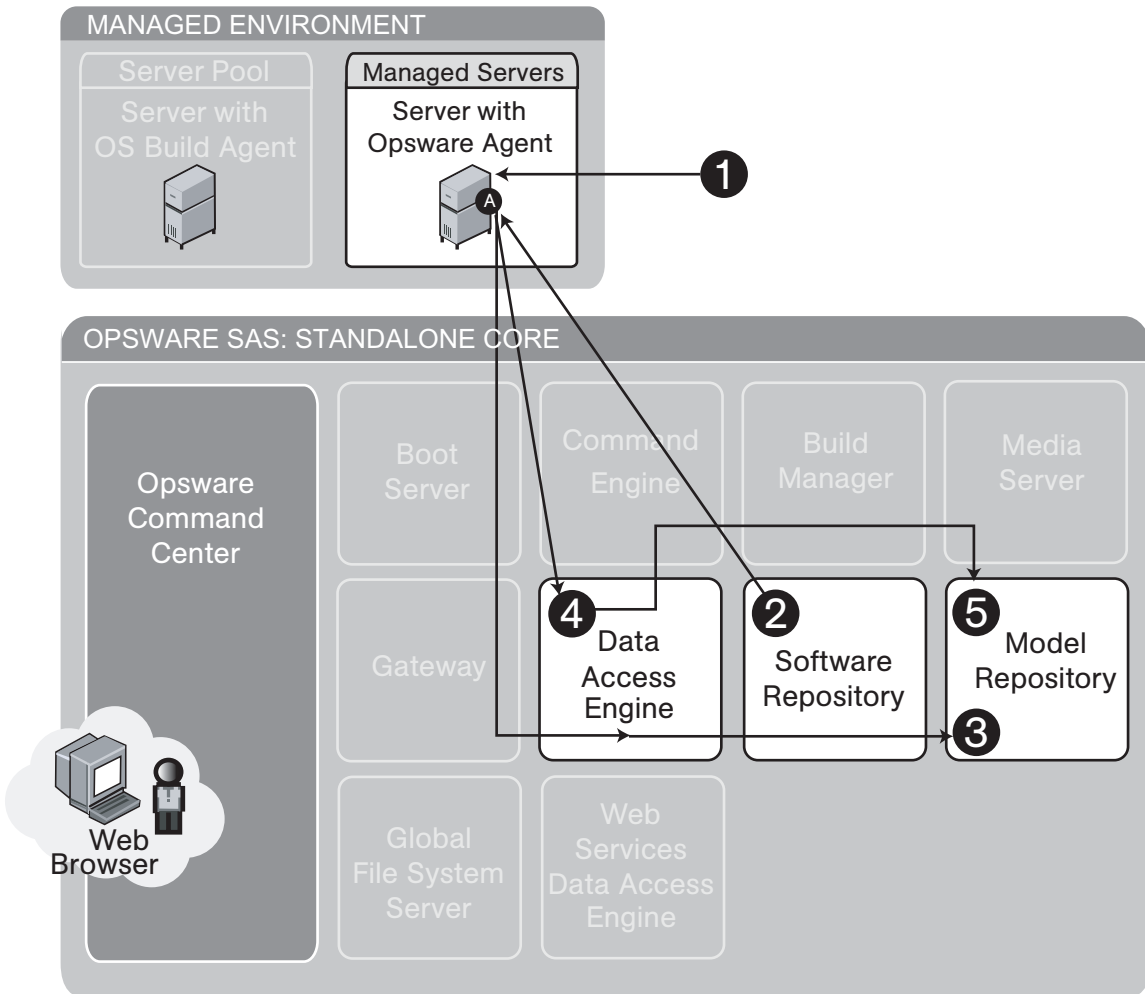


Figure 1-15 illustrates Opware SAS integration with AIX and HP-UX operating systems:

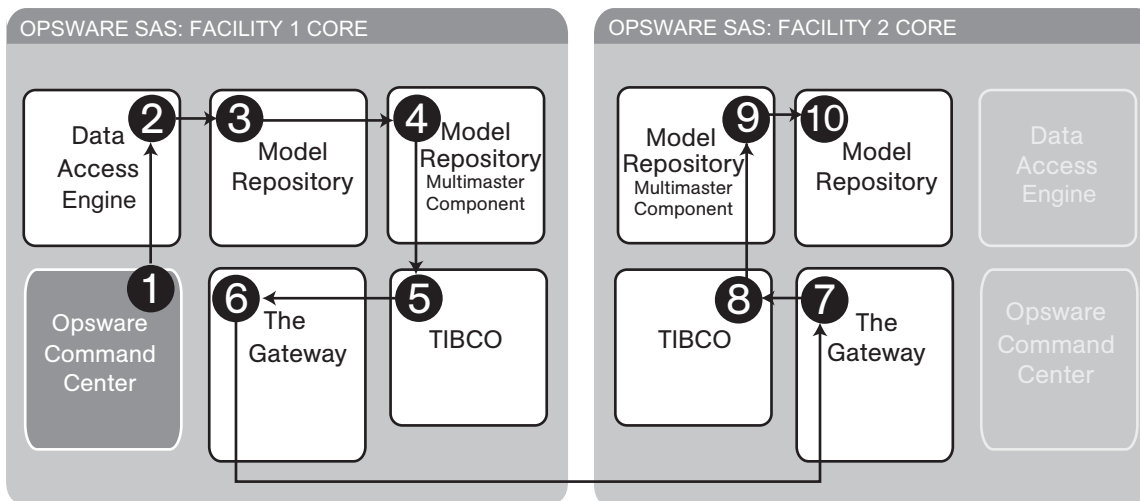
- 1** Installation technology installs the OS.
- 2** Opware SAS integration downloads and installs the Opware Agent on the server.

- 3** The Opware Agent determines hardware, software, customer, and facility information and records the server information in the Model Repository via the Data Access Engine.
- 4** The Opware Agent Installer attaches the server to the specified OS template.
- 5** (Optional) The server is reconciled with the modeled OS in the Model Repository.

Component Interaction in Multiple Facilities

Figure 1-16 shows how Opware SAS components interact when Opware SAS is running in multiple facilities. See “Overview of Multimaster Mesh Administration” on page 69 for information on how to administer this Opware SAS configuration.

Figure 1-16: Interaction Between Components in Multiple Facilities



- 1** An Opware user updates the managed environment.
- 2** The Data Access Engine sends an update to the Model Repository.
- 3** A trigger fires in the Model Repository, and the changes are saved in the transaction table in the Model Repository.
- 4** The Outbound Model Repository Multimaster Component monitors the transaction table for updates.
- 5** The Outbound Model Repository Multimaster Component publishes the updated message to TIBCO.
- 6** TIBCO connects to the Opware Gateway in Facility 1 and sends the updated message.

- 7** The updated message travels over the tunnel between facilities and arrives at the Opware Gateway in Facility 2.
- 8** The Opware Gateway in Facility 2 sends the message to TIBCO.
- 9** The Inbound Model Repository Multimaster Component in Facility 2 receives the TIBCO event with updates.
- 10** The Inbound Model Repository Multimaster Component in Facility 2 updates the local Model Repository.

Discovery and Agent Deployment

The Opware Discovery and Agent Deployment feature allows you to deploy Opware Agents to a large number of servers, enabling you to remotely deploy the Opware Agent to servers in your data center and place them under Opware management.

Figure 1-17: Interaction of Discovering Servers and Installing Agents

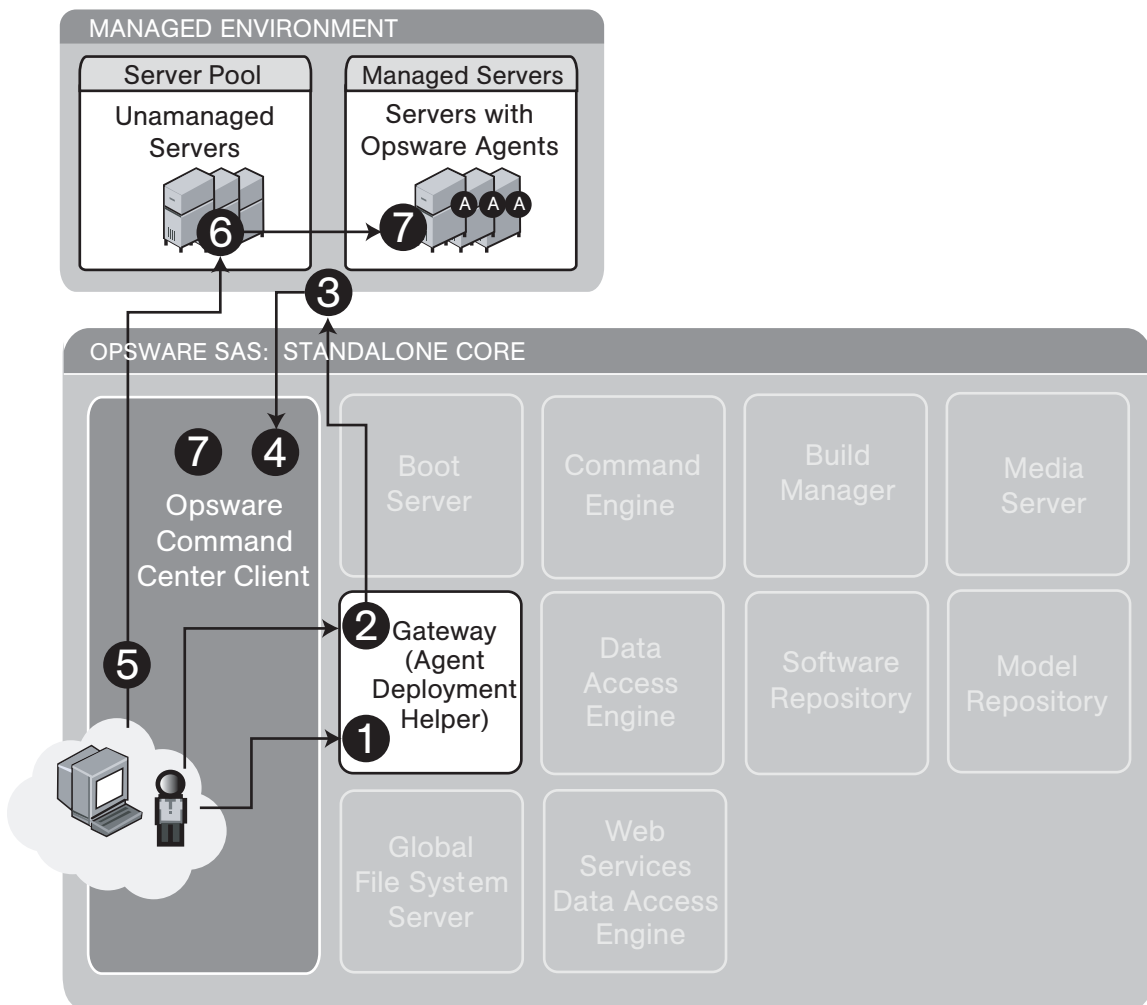


Figure 1-17 shows the process of discovering unmanaged servers and installing the Opware Agent on those servers:

- 1** An Opware user launches the Discover and Agent Deployment feature in the OCC Client and selects a scan location. Selecting a scan location selects the Agent Deployment Helper that will perform the scan. Each Opware Gateway is also an Agent Deployment Helper.

- 2** The user specifies a range of IP addresses to scan.
- 3** The Agent Deployment Helper scans those IPs, determines if anything is using those IP addresses and what ports are open.
- 4** Scan results are displayed in the OCC Client.
- 5** The user selects one or more servers, provides a login name and password, sets any install options and chooses the agent deployment option.

6 For Unix:

1. The Agent Deployment Helper tries to log onto the server by using available protocols.
2. It determines the operating system of the server.
3. It checks agent installation prerequisites.
4. It downloads the agent installer.
5. It installs the Opware Agent on the server.

For Windows:

1. The Windows Agent Deployment Helper establishes a tunnel via the Opware Gateway mesh to the server, then proceeds through the same steps as for Unix.
2. The list of servers is updated in the OCC Client to show the status of the Opware Agent installation.

Application Configuration Management

Opware Application Configuration Management (ACM) allows you to create configuration templates so you can modify and manage application configuration files associated with server applications. ACM enables you to manage and update and modify those configurations from a central location, so you can always be sure that applications in your data center are accurately and consistently configured the way you want them to be.

Figure 1-18: Application Configuration Management Process

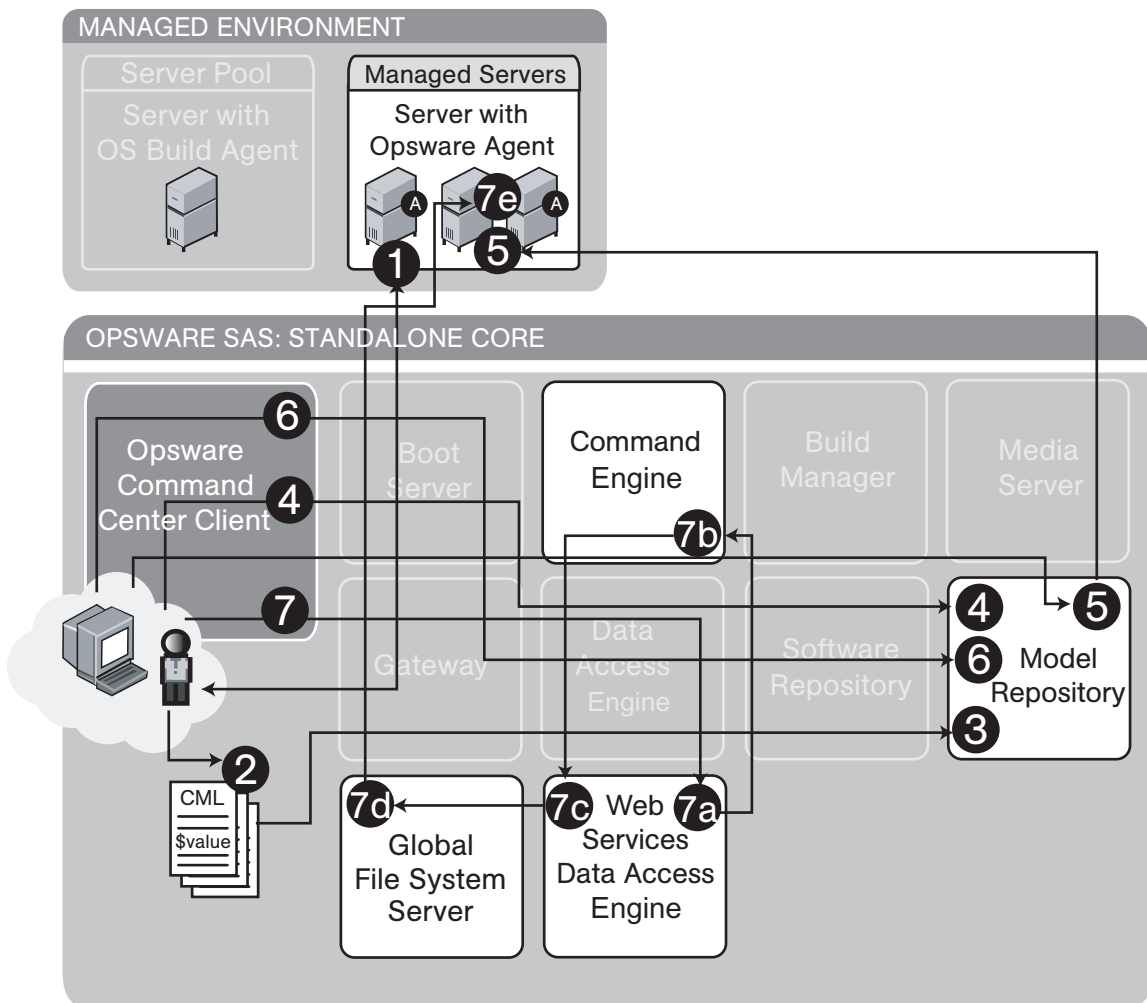


Figure 1-18 shows the process of discovering unmanaged servers and installing the Opware Agent on those servers:

Part A: Create an Application Configuration and Associated Templates

- 1** An Opware user chooses a “gold” configuration for an application on a managed server and retrieves the configuration files.
- 2** The user edits these configuration files, creating a CML file, turning some values into variables that can later be configured at a global or granular level.
- 3** The user creates templates for the Application Configuration and pastes in the edited CML files.
- 4** The user logs into the OCC Client and creates an Application Configuration, which is stored in the Model Repository.

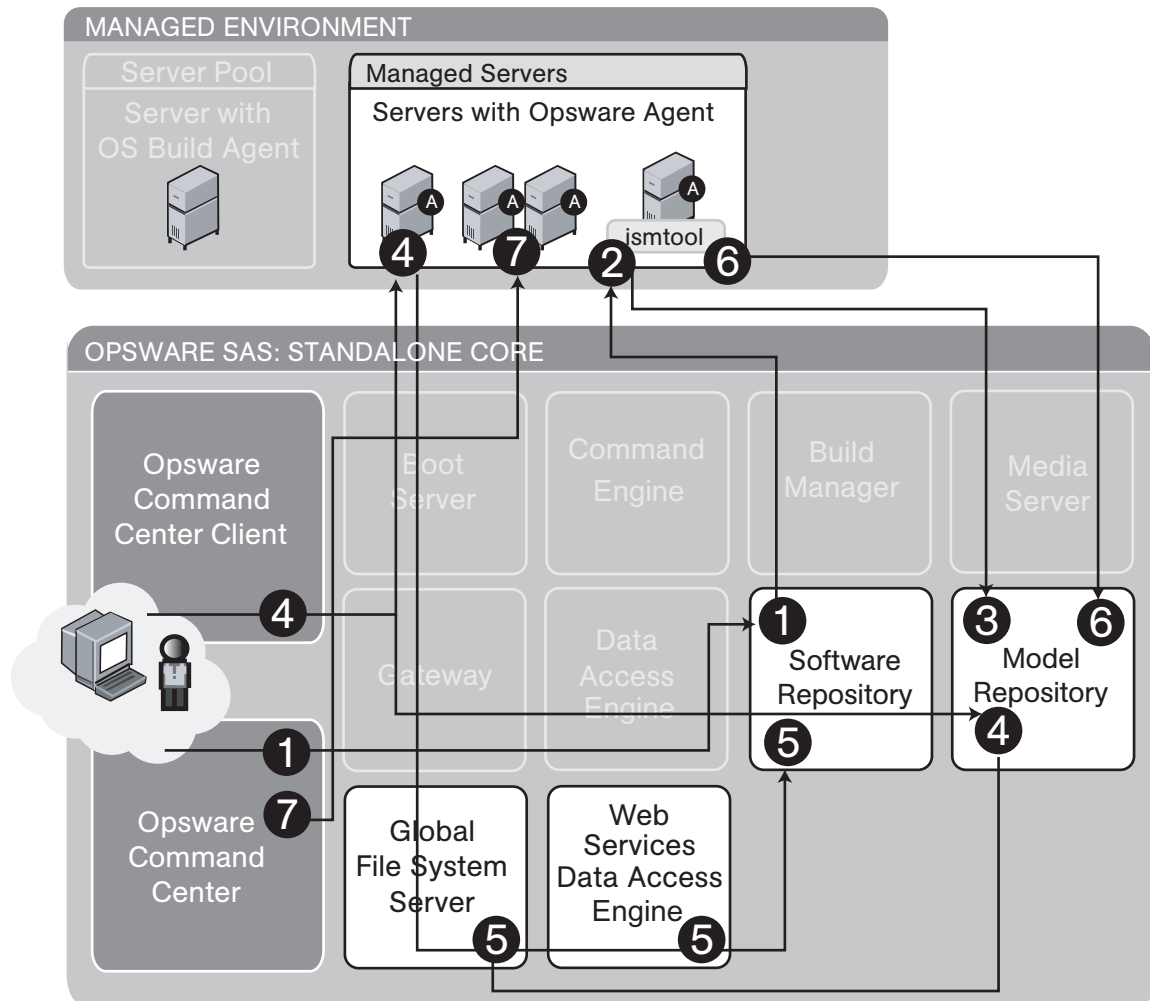
Part B: Configure and Push Application Configurations to Servers

- 5** The user chooses servers or server groups in the OCC Client and adds an Application Configuration to the target servers.
- 6** The user uses the Value Set Editor to configure the application for these servers, and these values are saved in the Model Repository.
- 7** The user clicks **Push** to enable the application configuration to the target servers. To accomplish this action, the Web Services Data Access Engine communicates with the Command Engine to create a session ID. The Command Engine then passes session data back to the Web Services Data Access Engine which communicates with the Global File System Server to push application configurations to managed servers.

Visual Packager

The Create Package feature allows you to create an installable software package based on compliance information, such as server snapshots and audit results. For each package, you can specify the customer assignment, reboot requirements, and pre/post install and pre/post uninstall scripts.

Figure 1-19: Interaction of Configuring the Packaging Server and Creating a Package



Part A: Configuring the Visual Packaging Server

- 1 An Opware user logs into the OCC and uses Managed Servers to select the server to use as a packaging server.

- 2 The user clicks Install ► By Template... to install the template.

Opware Tools ► Visual Packager ► [OS of server selected]

This will make the server a Visual Packaging server (by installing ismtool on it).

- 3 The Model Repository records this managed server as a visual packaging server for the chosen operating system.

This process is repeated for some or all of the operating systems in the environment.

- 4 The user then logs into the OCC Client and sets preferences to designate which managed servers to use as packaging servers for each operating system. Setting this preference queries the Model Repository via the Web Services Data Access Engine to figure out what packaging servers are available for the platform selected.

Part B: Creating and Installing a Visual Package

- 4 The user logs into the OCC Client and selects one of the following items as a source for the visual package:

- A Server (from managed servers)
- An Audit Result (from the Model Repository)
- A Snapshot Result (from the Model Repository)

- 5 The user selects package contents and creates a visual package, which is stored in the Software Repository.

File system and registry resources are accessed via the Global File System Server. All other operations go to the Web Services Data Access Engine.

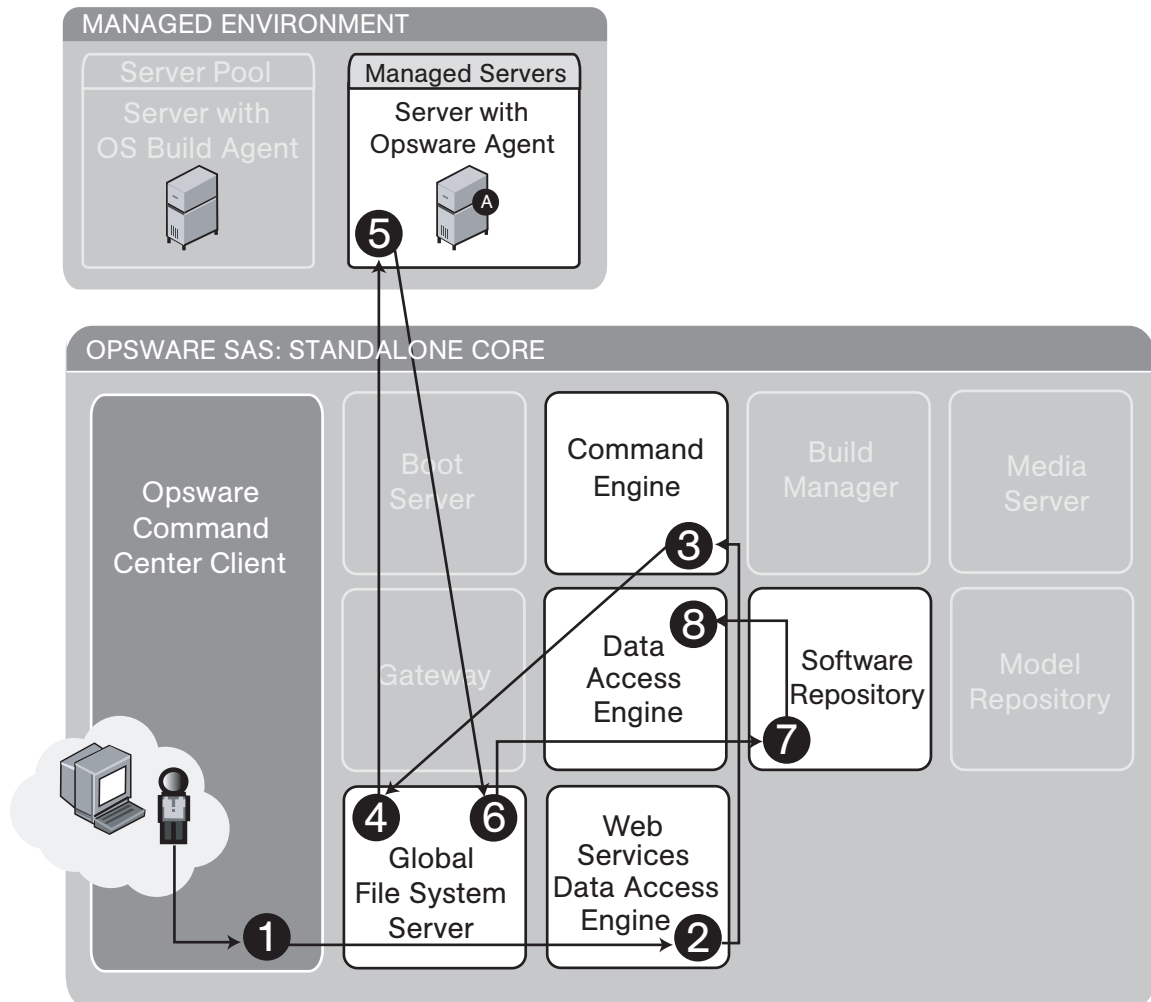
- 6 The ismtool creates a node in the Software Tree on the model and attaches the new package.

- 7 The user logs into the Opware Command Center (web client) and installs the new visual package onto managed servers.

Server Compliance and Auditing

The Opware Server Compliance and Auditing feature enables Opware users to keep managed servers up-to-date by comparing them to known working servers.

Figure 1-20: Component Interaction of Taking snapshots

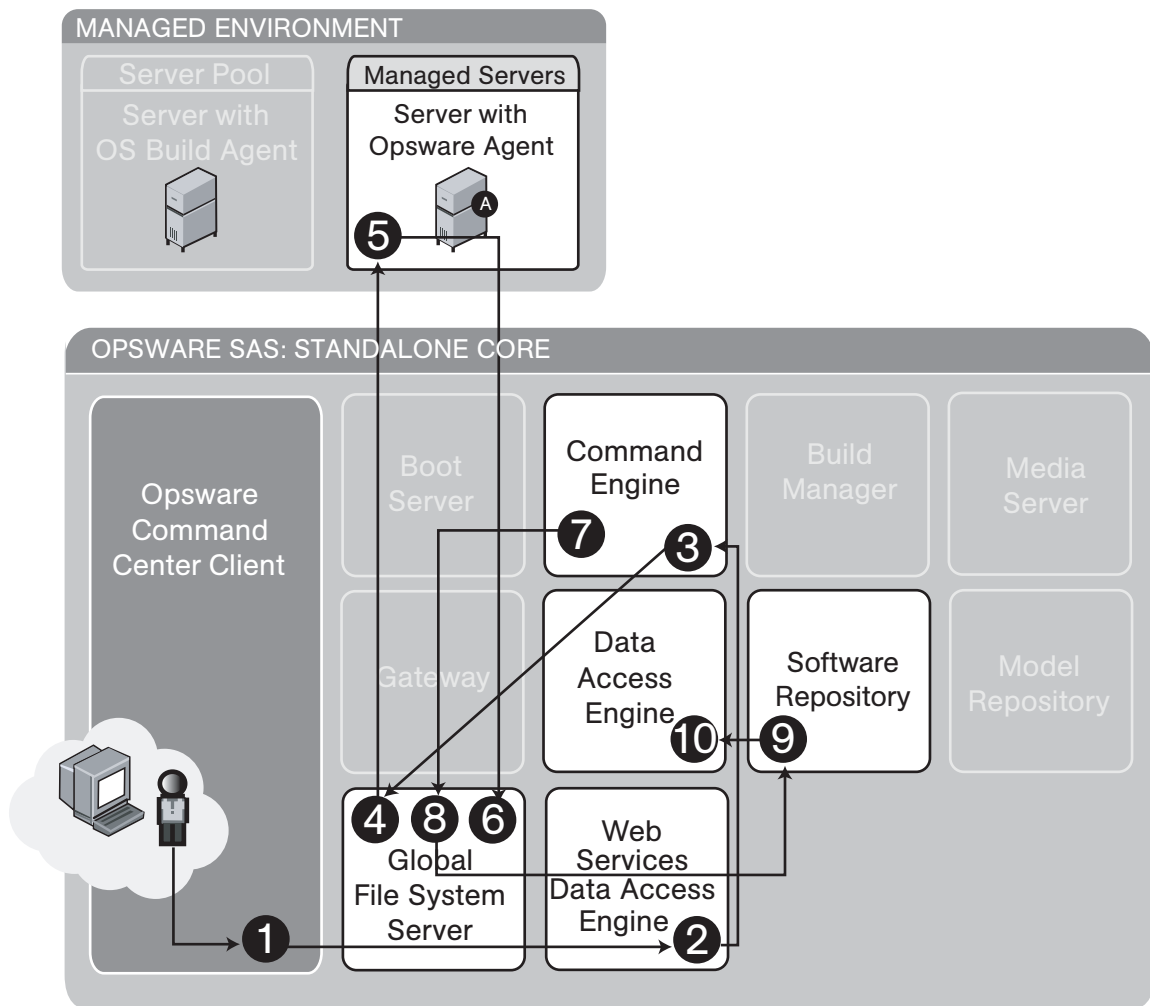


Server Compliance: Take a Snapshot

- 1** An Opware user chooses a Snapshot template to use.
- 2** The user clicks **Run**, which invokes the appropriate command on the Web Services Data Access Engine.

- 3** The Web Services Data Access Engine communicates with the Command Engine to coordinate the snapshot.
- 4** The Global File System Server is used to provide snapshot information from the managed server.
- 5** The snapshot information is assembled in the Global File System Server.
- 6** The snapshot information recorded is stored in the Software Repository.
- 7** The snapshot information is stored in the Data Access Engine.

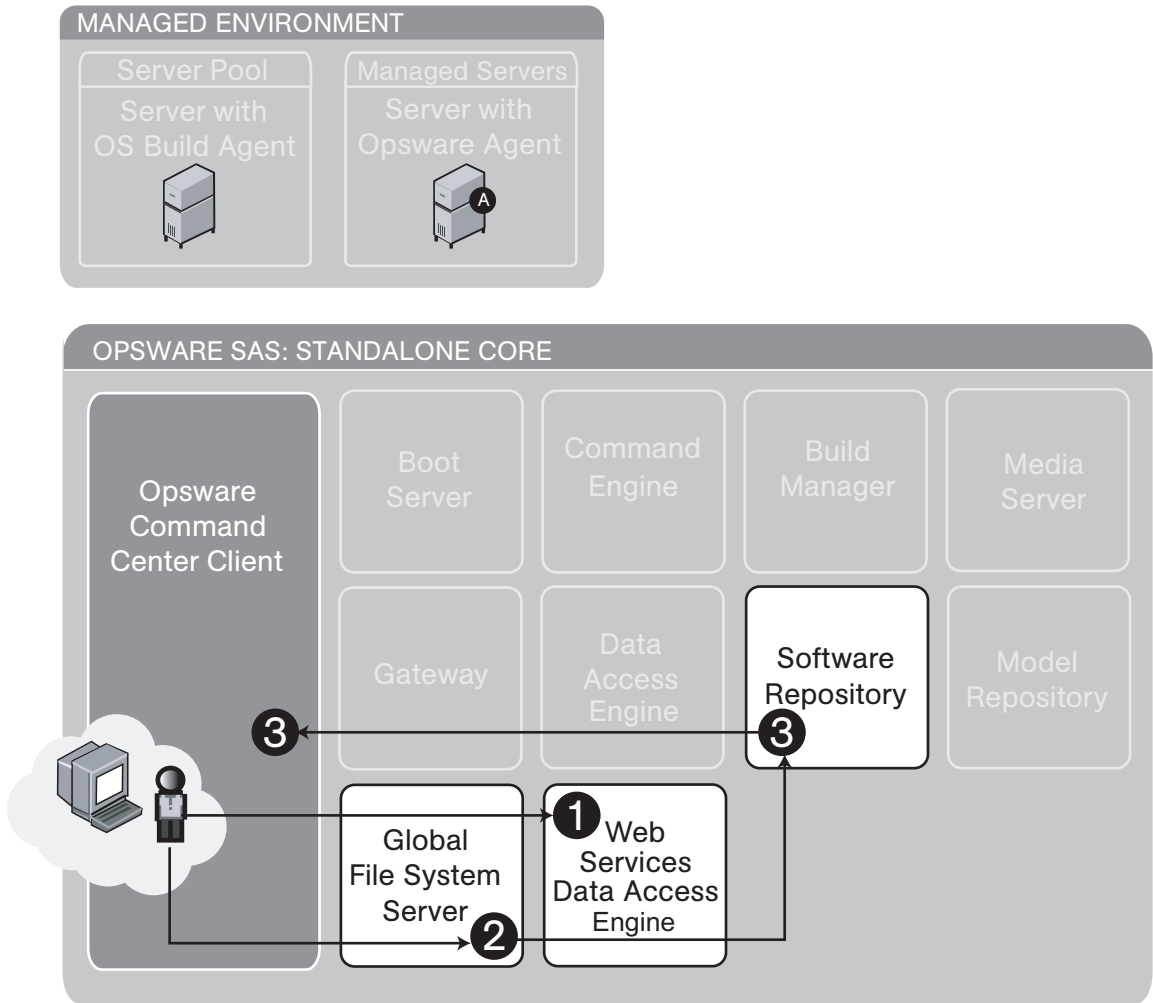
Figure 1-21: Component Interaction of Auditing a Server



Server Compliance: Run an Audit

- 1** An Opware user chooses an audit template to use.
- 2** The user clicks **Run**, which invokes the appropriate command on the Web Services Data Access Engine.
- 3** The Web Services Data Access Engine communicates with the Command Engine to coordinate the audit.
- 4** The Global File System Server is used to provide audit information from the managed server.
- 5** The audit information is assembled in the Global File System Server.
- 6** The Command Engine issues the create audit command.
- 7** The Global File System Server loads appropriate snapshots and performs a difference.
- 8** The resulting audit is uploaded to the Software Repository.
- 9** The audit is stored in the Data Access Engine.

Figure 1-22: Component Interaction of Viewing Snapshot or Audit Results



Server Compliance: View results of audit or snapshot

- 1** An Opware user gets a list of available snapshots or audit results information.
- 2** The user requests detailed information about a snapshot or an audit.
- 3** The results are returned from the Software Repository to the user.

Chapter 2: Opware Multimaster Mesh Administration

IN THIS CHAPTER

This section discusses the following topics:

- Overview of Opware Multimaster Mesh
- Multimaster Facilities Administration
- Multimaster Mesh Administration
- Best Practices for Preventing Multimaster Conflicts
- Examining the State of the Multimaster Mesh
- Best Practices for Resolving Database Conflicts
- Model Repository Multimaster Component Conflicts

Overview of Opware Multimaster Mesh



This guide does not document how to set up Opware SAS to run in a multimaster mesh. For more information, see the *Opware[®] SAS Deployment and Installation Guide* or consult your Opware SAS Support Representative.

A multimaster mesh is a set of Opware cores with synchronized Model Repositories. A multimaster mesh has the following characteristics:

- Each core is associated with a specific facility.
- Each facility is independent of the other facilities.
- The Model Repositories in the different facilities are geographically dispersed.
- Data is updated locally and then propagated to every Model Repository in the multimaster mesh.

- The Model Repositories are available for both read and write transactions.
- The multimaster mesh is invisible to operations personnel.

Running Opware SAS in a multimaster mesh has the following advantages:

- **Redundancy:** If a core in one facility becomes unavailable, the Opware Command Center is still usable from other facilities. Users in other facilities can have their own Opware Command Center. Also, it provides the ability to move out of a facility and keep Opware SAS running in other facilities.
- **Performance scalability:** Write operations do not need to be proxied to a central location.
- **Geographic scaling:** International facilities can be independent and do not need to rely on a network connection across continents to a central facility.

Multimaster Facilities Administration

In the Opware Command Center, a facility refers to the collection of servers that a single Opware core or Satellite manages. A facility can be all or part of a data center, server room, or computer lab. Users can manage servers in any facility from the Opware Command Center in any facility. When a user updates data in a facility, the Model Repository for that facility is synchronized with the Model Repository databases located in all remote facilities. In the Opware Command Center, a facility is identified by a facility name and a facility ID.

Updating Facility Information and Settings

Perform the following steps to update facility information and settings:

- 1** From the navigation panel, click Environment ► Facilities. The Facilities page appears and displays the names of the current facilities.

- Click the hyperlink name of the facility that you want to update. The Facilities: Edit Facility page appears with the Properties tab automatically selected, as Figure 2-1 shows.

Figure 2-1: Properties Tab of the Edit Facility Page

Facilities: Edit Facility

Return to Facilities

Properties		Custom Attributes
Facility Information		
Facility ID:	3	
Name:	<input type="text" value="DATACENTER1"/>	
Short Name:	TR3	
Is this facility in use?	Yes	
Customers:	<input type="checkbox"/> Customer Independent <input type="checkbox"/> MYCUSTOMER	
		<input type="button" value="Save"/> <input type="button" value="Cancel"/>

- To change the name of the facility that appears in the Opsware Command Center, edit the Name field or click the Return to Facilities link to exit without making any changes.



Contact your Opsware SAS Support Representative if you need to make other changes to the facility properties

- Click **Save**. The Opsware Command Center displays a message that confirms that the properties for that facility were updated.
- Select the Custom Attributes tab.

The Custom Attributes page appears, which provides name-value pairs associated with this customer. These named values are used to provide parameters to Opware SAS, for example, to customize displays or provide settings to use during installation or configuration of packaged software in the operational environment.

- 6** Click the hyperlinked name of an attribute to display the Facilities: Edit Attribute for [facility name] and make changes to its associated value.
- 7** To add an attribute name and to specify a value to associate with the attribute, click **New**.



Be careful when you update or remove existing attribute settings as it might affect or disrupt operation of the operational environment. Contact your Opware SAS Support Representative to help you determine the appropriate changes to make when you update the information or settings for a specific facility.

- 8** When you finish making updates to the facility properties or custom attributes, click the Return to Facilities link.

Multimaster Mesh Administration

This section provides information on multimaster mesh administration within Opware SAS and contains the following topics:

- Overview of Multimaster Mesh Administration
- Model Repository Multimaster Component Conflicts
- Causes of Conflicts
- User Overlap
- User Duplication of Actions
- Connectivity Problems that Cause Out of Order Transactions

Overview of Multimaster Mesh Administration

Multimaster is a configuration for synchronizing copies of the Model Repository database located in different facilities. Any Model Repository database in any facility can be used as the updateable source at any time. In the multimaster architecture, there is no designated master for any individual data element.

Operating in a multimaster architecture involves the chance of conflicting updates being made to the same record in different Model Repository databases. The multimaster Opware SAS components detect conflicts and propagate alerts; however, multimaster components do not resolve conflicts. Opware administrators use the multimaster tools in the Opware Command Center to resolve the conflicts at the target databases.

In a configuration with multiple facilities, one facility, known as multimaster central, is designated as the primary facility. It is responsible for generating the transaction table automatically, although the other facilities can also do so upon demand. Multimaster central is automatically defined during installation. See “Designating the Multimaster Central Data Access Engine” on page 150 in Chapter 4 for more information

Model Repository Multimaster Component Conflicts

When an update from the source database arrives at the destination database, a conflict can be generated any time the data at the destination database is not what was expected – either the values are different or the row cannot be found.

The probability of multimaster conflicts occurring varies depending on the following factors:

- The number of servers under management
- The number of facilities
- The number of Opware Command Centers
- The propensity for users to make changes in more than one facility by using different Opware Command Centers

Data conflicts occur when the values of the objects in the local Model Repository do not match the values in a message from the Outbound Model Repository Multimaster Component or a database constraint is violated.

When a conflict is flagged, Opware SAS takes the following actions:

- 1** The transaction is canceled.
- 2** All rows affected by the transaction are locked, thereby preventing further changes to those rows.
- 3** The Outbound Model Repository Multimaster Component propagates this change in a new transaction to all remote databases, thereby locking the rows in all facilities.
- 4** An alert message with the conflict information is emailed to the configured mailing list.
- 5** The Inbound Model Repository Multimaster Component continues on to the next message.

If the Inbound or Outbound Model Repository Multimaster Component encounters an exception that prevents it from going on to the next message, it sends an email and shuts itself down.



An Opware administrator must manually resolve the problem by using the Opware Command Center. Resolving the conflict unlocks the rows. See “Best Practices for Resolving Database Conflicts” on page 74 in this chapter for more information

Causes of Conflicts

Conflicts can have the following causes:

- User Overlap
- User Duplication of Actions
- Connectivity Problems that Cause Out of Order Transactions

User Overlap

Multiple users are working in the same area of data by using the Opware Command Centers in different facilities. Conflicts occur when a user makes a change by using the Opware Command Center in one facility and another user makes a change to the same object using the Opware Command Center in another facility.

Partitioning the data space helps to reduce the number of conflicts that user overlap causes.

For example, this sequence of events occurs:

- 1** Alice removes Node A from a server in the Atlanta facility.
- 2** Bob removes Node A from the same server in the Boston facility.
- 3** Opware SAS propagates the change from the Atlanta facility to the Boston facility; however, the node has already been removed from the server in the Boston facility. Opware SAS generates a Model Repository Multimaster Component conflict.
- 4** Opware SAS propagates the change from the Boston facility to the Atlanta facility; however, the node has already been removed from the server in the Atlanta facility. Opware SAS generates a second Model Repository Multimaster Component conflict.

User Duplication of Actions

Conflicts occur when a user makes a change in one database, does not see the change reflected in another database, and makes the change again in the other database.

This situation involves a user bouncing back and forth between multiple Opware Command Centers, or between an Opware Command Center and some command line utilities in a facility.

For example, this sequence of events occurs:

- 1** From a server in the Seattle facility, Carol uses the Opware Command Line Interface (OCLI) to upload the package `carol.conf`.
- 2** In the Phoenix facility, Carol logs into the Opware Command Center to search for the package. She does not see the package because that data has not yet propagated from Seattle to Phoenix. Carol is unaware of the lag time for data propagation between facilities.
- 3** Carol uploads the package `carol.conf` by using the Opware Command Center in Phoenix.

When the data arrives from Seattle, Opware SAS generates a conflict because the data already exists in Phoenix.

Connectivity Problems that Cause Out of Order Transactions

This situation causes conflicts when a user changes or inserts data at facility A (Model Repository database A). The transaction for that change propagates to facility B (Model Repository database B). The same data is modified again or somehow referenced at facility B (Model Repository database B). The transaction from facility B reaches facility C (Model Repository database C) before the transaction from facility A.

Transactions sent from a facility to another facility arrive in the order in which they were sent. However, the correct ordering is not guaranteed for transactions arriving from different facilities.

This type of conflict occurs only when Opware SAS is running from three or more facilities.

A common cause of this situation is a user uploading a package by using the OCLI, and then immediately adding the package to a node by using the Opware Command Center in another facility. The delay in propagating data about the package to other facilities causes the data about the node attachments to arrive at other facilities out of order.

The occurrence of out of order transactions is aggravated by proximate updates in different facilities and unreliable inter-facility network connections.

For example, this sequence of events occurs:

- 1** From a server in the Denver facility, Henry uses the OCLI to upload the package `henry.conf`.
- 2** Opware SAS propagates data about the package to the Miami facility; however, it cannot propagate the data to the Paris facility because the network connection to the facility is down.
- 3** Henry updates the description of the package `henry.conf` by using the Opware Command Center in Miami.
- 4** Opware SAS propagates data about the updated package description to the Denver facility; however, it cannot propagate the data to the Paris facility because the network connection to the facility is down.
- 5** Network connectivity to the Paris facility is restored and multimaster messages are propagated to the Paris facility.
- 6** The message about the updated package description arrives at the Paris facility before the message about the uploaded package. The Model Repository in the Paris facility does not contain data about the package, so a conflict is generated.

- 7 The message about the uploaded package arrives at the Paris facility and is processed without error. The package data exists in Paris but the package description differs from the other facilities.

Best Practices for Preventing Multimaster Conflicts

When you use Opware SAS in multiple facilities, try to keep the number of conflicts that can occur to a minimum. Educate users to consider the following factors when Opware SAS is running in a multimaster mesh:

- Users in multiple facilities are able to modify the same data at the same time.
- A slight time delay occurs before changes that a user makes arrive in other Opware SAS facilities. (The length of delay varies depending on a number of factors, including network connectivity and bandwidth.)

Implement these best practices to reduce the chance of data conflicts between facilities:

- Ensure reliable network connections and sufficient network bandwidth between facilities. The risk of conflicts increases with degraded network connectivity between facilities.

See “Network Administration for Multimaster” on page 88 in this chapter for more information.

For additional assistance, consult your Opware SAS Support Representative or see the *Opware[®] SAS Deployment and Installation Guide* for information about network connectivity when running Opware SAS with a multimaster mesh.

- Educate users not to change data in one facility and then make the same change in another facility.
- Partition the data space so that more than one user does *not* change the same object in different facilities at the same time.

Have a user or a small group of coordinated users manage a given set of servers. Partitioning the data space ensures accountability of server ownership and prevents users from changing each other's data.

Opware SAS includes a mechanism for distributed access to data. Specifically, the Opware Command Center includes permissions by customer, facility, and User Group Types.

See the *Opware® SAS Configuration Guide* for more information about User Groups and Opware SAS Permissions.

Examining the State of the Multimaster Mesh

You can examine the state of the multimaster mesh by clicking the Multimaster Tools option, which is visible in the Opware Command Center at all multiple facility installations.

When you select the Multimaster Tools option, the Multimaster Tools: State View page appears. In addition to a color-coded legend that shows possible transaction states (including red for Conflict, orange for Not Sent, yellow for Not Received, Gray for Unable to Connect, and green for Good), this page also:

- Presents an overview of the health of the multimaster mesh by automatically checking all facilities.
- Shows the state of the last five transactions – a unit of change to a database that consists of one or more updates to rows and has a globally unique transaction ID – from each facility to each other facility and also shows all conflicting and all unpublished transactions.
- Shows the time that the Opware Command Center generated and cached the data. Click **Refresh** to refresh that cached data.

Opware administrators can also use the System Diagnosis tools in the Opware Command Center to view information about the health of the multimaster components.

See “Opware SAS Diagnosis” on page 121 in Chapter 4 for more information.

Best Practices for Resolving Database Conflicts

Maintaining data consistency is complex and conflicts can occur even when implementation and work processes minimize them. This section contains the following topics:

- Types of Conflicts
- Guidelines for Resolving Each Type of Conflict

Types of Conflicts

The following types of conflicts can occur:

- **Identical data conflict:** The Multimaster Tools show a conflicting transaction but the data is the same between facilities. The data is the same because users made the same change in different facilities.
- **Simple transaction conflict:** The row exists in all facilities, but some columns have different values or the row does *not* exist in some facilities (missing objects).
- **Unique-key constraint conflict:** The object does not exist in a facility and cannot be inserted there because inserting it would violate a unique-key constraint.
- **Foreign-key constraint conflict:** The row does not exist in some facilities and cannot be inserted because the data contains a foreign key to another object that also does not exist in that facility.
- **Linked object conflict:** A type of conflict encountered in rare cases. Opware SAS includes business logic that links specific related objects in Opware SAS, such as a custom attribute name and value, and a customer created in the Opware Command Center UI (appears in lists) and the associated node for the customer in the node hierarchy. Opware SAS ensures that links between related objects are maintained. Resolving a linked object conflict can be complex because you must attempt to preserve the intent of the transaction that caused the conflict. Contact your Opware SAS Support Representative to help you resolve linked object conflicts.

Guidelines for Resolving Each Type of Conflict

In general, when you resolve conflicts, apply updates so that the target always reflects the most current data based on the time stamp of the originating changes.

When you cannot follow one of the preceding guidelines, attempt to preserve the intent of the transaction. Contact the users who are generating the transactions and determine what types of changes in the managed environment each user was trying to make.

Identical Data Conflict

All objects in a transaction contain exactly the same data across all facilities. This type of conflict includes the case where the objects do not exist in all facilities.

To resolve an identical data conflict, simply mark the conflict resolved.

Identical Data Conflict (Locked)

All objects in a transaction contain exactly the same data across all facilities but the objects in the transaction are still locked (marked conflicting).

To resolve this type of conflict, pick an arbitrary facility and synchronize all objects from it. Performing this action unlocks the objects. After synchronizing the data, mark the conflict resolved.

Simple Transaction Conflict

The data is different between facilities or some objects are missing from some facilities. None of the objects depend on the actions of other conflicting transactions. The results of synchronizing the objects does not result in a database foreign-key or unique-key constraint violation.

To resolve a simple transaction conflict, choose the facility that contains the correct data and synchronize from it. How you determine which facility contains the correct data varies depending on the type of transaction:

- If the conflict is the result of two users overriding each other's work, talk to the users and determine which user's change should be correct.
- If the conflict is the result of automated processes overriding each other's data, the most recent change is usually correct.
- If the conflict is the result of out-of-order transactions, the most recent change is usually correct.

After synchronizing the data, mark the conflict resolved.

Unique-Key Constraint Conflict

Resolving these conflicts results in a unique-key constraint violation.

For example, this sequence of events occurs:

- 1** From the Opware Command Center in the London facility, John creates Node A1 as a subordinate node of Node A.
- 2** From the Opware Command Center in the San Francisco facility, Ann performs the same action. She creates Node A1 as a subordinate node of Node A.
- 3** Node names must be unique in each branch of the node hierarchy.
- 4** Opware SAS propagates the node changes from the London and San Francisco facilities to the other facilities. Inserting the rows into the Model Repository databases at other facilities causes a unique-key constraint violation and a conflict.

Resolving this conflict by inserting the updates from the London facility in all facilities would fail with the same unique-key constraint violation.

Perform the following steps to resolve a unique-key constraint conflict:

- 1** Locate all the involved transactions and synchronize one transaction from a facility where the object does not exist, thereby deleting it in all facilities.
- 2** Synchronize the other transaction from a facility where the object exists, thereby inserting the object in all facilities. One of the two uniquely conflicting objects will take the place of the other.

Foreign-Key Constraint Conflict

Resolving these conflicts results in a foreign-key constraint violation.

For example, this sequence of events occurs:

- 1** Jerry creates Node B in facility 1.
- 2** Before that transaction has time to propagate to other facilities, Jerry creates Node C as a subordinate node of Node B.
- 3** When the first transaction arrives at facility 2, it generates a conflict for unrelated reasons.
- 4** When the second transaction arrives at facility 2, inserting the row for Node C causes a foreign-key constraint conflict because the parent Node (Node B) does not exist.

Resolving the second conflict first by inserting the update for Node C into all facilities would fail with the same foreign-key constraint violation.

Perform the following steps to resolve a foreign-key constraint conflict:

- 1** Resolve the conflicting transaction for Node B (the parent Node) by synchronizing the first transaction from the facility where the object exists.
- 2** Synchronize the second transaction (the Node C update) from the facility where the object exists.

Generally, resolving conflicts in the order in which they were created avoids generating foreign-key constraint conflicts.

Model Repository Multimaster Component Conflicts

This section provides information on resolving model repository, multimaster component conflicts and contains the following topics:

- Overview of Resolving Model Repository Multimaster Component Conflicts
- Resolving a Conflict by Object
- Resolving a Conflict by Transaction

Overview of Resolving Model Repository Multimaster Component Conflicts

Opware administrators can view and resolve multimaster conflicts in any Opware Command Center by using the Multimaster Tools. The Multimaster Tools are available in all Opware Command Centers.



Before you resolve conflicts, notify the subscribers of the email alert alias. Notifying these users helps to prevent other Opware administrators from undoing or affecting each other's conflict resolution efforts. While resolving conflicts, you should resolve the conflict from the Opware Command Center of a single facility. Do not attempt to resolve the same conflict multiple times from the OCC of different facilities.



If you see a large volume of conflicts that you cannot resolve by using the Multimaster Tools, contact your Opware SAS Support Representative for assistance synchronizing databases.

Resolving a Conflict by Object

Perform the following steps to resolve conflicting transactions by object:

- 1 From the navigation panel, click Administration ► Multimaster Tools. The Multimaster Tools: State View page appears, showing a summary of all transactions and, if they exist, all conflicts. See Figure 2-2.

Figure 2-2: Transaction Table That Shows Conflicts

Multimaster Tools : State View			
State View		Conflict View	
Refresh			
Key			
Problem	Potential Problem		Good
<input checked="" type="checkbox"/> Conflict	<input type="checkbox"/> Not Sent	<input type="checkbox"/> Not Received	<input type="checkbox"/> Unable To Connect
			<input checked="" type="checkbox"/> Received
Transaction Status Counts			
		SOURCE FACILITY	
		C33	C34
D E S T I N A T I O N	C33		<input checked="" type="checkbox"/> 5 <input checked="" type="checkbox"/> 1
	C34	<input checked="" type="checkbox"/> 5 <input checked="" type="checkbox"/> 2	
F A C I L I T Y			


Generated: 10/28/04 10:39:43

Different types of transaction statuses are indicated by color-coded boxes:

- **Green:** The last five transactions that were successfully sent.
- **Orange:** All transactions that have not been published (sent to other facilities).
- **Red:** All conflicts.

Each box is displayed in a color scheme to indicate the status and success of the transaction. A key that explains the significance of the colors, like the one shown in Figure 2-3, is listed at the top of the page.


Figure 2-3: Conflict Color Key

Key				
Problem	Potential Problem			Good
 Conflict	 Not Sent	 Not Received	 Unable To Connect	 Received

Red boxes indicate that one or more transactions between facilities are in conflict and need to be resolved.

- To resolve a conflict, select the Conflict View tab. The Multimaster Tools: Conflict View page appears, as shown in Figure 2-4.

Figure 2-4: Transaction Differences Page That Lists all Transactions In Conflict in the Multimaster Mesh

Multimaster Tools : Conflict View 							
State View		Conflict View					
Refresh							
Transaction	Action	Table	Count	User	Published (UTC)	Source Facility	Conflicting
566530001	Update	DEVICE_CHANGE_LOG	4	ROOT	10/28/04 10:29:33	C33	C34
566560001	Update	DEVICE_CHANGE_LOG	4	ROOT	10/28/04 10:29:33	C33	C34
514380002	Update	DEVICE_CHANGE_LOG	4	ROOT	10/28/04 10:29:41	C34	C33
Generated: 10/28/04 10:30:22							

The page lists each transaction by ID number (clickable link), the actions that caused the conflict, the database objects affected by the conflict, the user responsible for the conflict (listed by the IP of the OCC where the user made the change), when the offending action occurred, the source facility that originated the transaction, and the facilities where the transaction conflicted.



The page might show a conflict where the data is the same in both facilities but a conflict exists, because the same change was made in both facilities. Even though the data is correct, the conflict still exists and must be resolved. See “Best Practices for Resolving Database Conflicts” on page 74 in this chapter for more information

- 3 To resolve a conflict, click the transaction ID number link. You see the Multimaster Tools: Transaction Differences page, which shows a comparison of the objects between facilities, with any differences shown in red, as illustrated in Figure 2-5.

Figure 2-5: Transaction Differences Page for Multimaster Tools Showing Conflicts Between Facilities


Multimaster Tools: Transaction Differences 566530001 from Source Facility C33		
Return to Conflict View		
Synchronize all objects from C34 <input type="button" value="Update"/>		
DeviceChangeLog 440001		
DB Field	C34	C33
CHANGE_SUMMARY	SZXT3Aip fKk ZMv2clBBE2pN2LdXSikB0GgKwdqG2 VbM7klQ R awY s6T X oXPvRpRjqw HdRGgJPLg Bh CP7sSGGJfS1	h1V mflbM3lw IHQqj4i fd h nLB4 L044IK7Dg9qoYLK5wQkFnSgik J645XZYMjc wP FEMvhufpBIUqv5fONOB VTKzcp
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	440001	440001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566590001
	<input type="button" value="Synchronize From"/>	<input type="button" value="Synchronize From"/>
DeviceChangeLog 450001		
DB Field	C34	C33
CHANGE_SUMMARY	78vzxGYnqIcBggfjD StA1VU3LZkBSyY4 M NRJfPPRZyL WxVaiNAr P0OtheHMnLHMRA nX lh J 1kGIK zMLR8l Yh YrFI	QuuM YPFNFH2cT 0wspWxvPZDGL9doTSvm9L8F z FIZ8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu LxKq
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	450001	450001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566590001
	<input type="button" value="Synchronize From"/>	<input type="button" value="Synchronize From"/>
DeviceChangeLog 460001		
DB Field	C34	C33
CHANGE_SUMMARY	e M mwJWim6xPHM9nB0u mGOGX0 HPgQB4438szTrguhK2P11w A49w2 JE7QG99vuiznC rwC1ysjeB P sXsWrtZ8dx	OJzpb6C K FXueIN8PcJg3KFe7 juKiaqTIVoTAEMdtiV0sA1Ew4ZPAwV c MxB0VxREERDH yV w6Ryf 7l v pX
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	460001	460001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566590001
	<input type="button" value="Synchronize From"/>	<input type="button" value="Synchronize From"/>

- 4 To resolve each object, click **Synchronize From** at the bottom of the object.

The Multimaster Tools insert or delete objects in the transaction where necessary, and then propagate the change to every facility in the multimaster mesh.

The Multimaster Tools: Object Synchronization Results page appears, displaying the results of the transaction synchronization, as shown in Figure 2-6.

Figure 2-6: Object Synchronization Result Page

Multimaster Tools: Object Synchronization Result | DeviceChangeLog 440001 


Return to Transaction Differences

Object successfully synchronized.

Table	Facility	Action
DeviceChangeLog 440001	C34	Unlock
	C33	Update

- Click the Return to Transaction Differences link. The Multimaster Tools: Transaction Difference page appears. Notice that the object you synchronized shows on the page as being identical between the facilities, as shown in Figure 2-7.

Figure 2-7: Single Object Resolved

Multimaster Tools: Transaction Differences | 566530001 from Source Facility C33 

[Return to Conflict View](#)

Synchronize all objects from C34

DeviceChangeLog 440001		
DB Field	C34	C33
CHANGE_SUMMARY	SZXT3Aip fKk ZMv2cIBBe2pN2LdXSikB0GgKwdqG2 Vbm7kQ R aWY s6T X oIXPvRpRjqw HdRGgJPLg Bh CP7sSGgfS1	SZXT3Aip fKk ZMv2cIBBe2pN2LdXSikB0GgKwdqG2 Vbm7kQ R aWY s6T X oIXPvRpRjqw HdRGgJPLg Bh CP7sSGgfS1
CONFLICTING	0	0
DVC_CHANGE_LOG_ID	440001	440001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566510001

DeviceChangeLog 450001		
DB Field	C34	C33
CHANGE_SUMMARY	78vzGYNqICbgqfjD StA1VU3LZkBSyY4 M NRJfPPRZyL WxValINAr POOtheHMnLHMRA nX Ih J 1kQIK zMLr8l Yh YrFi	QuuM YPFNFH2cT 0wspWxvPZDGL9doTSvm9L8F z Fz8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu LxKq
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	450001	450001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566590001
	<input type="button" value="Synchronize From"/>	<input type="button" value="Synchronize From"/>

DeviceChangeLog 460001		
DB Field	C34	C33
CHANGE_SUMMARY	e M mwJWim6xPHM9nB0u mGOGX0 HPgQB443SzTrguhkr2P1t w A49w2 jE7QG99vuznC rwC1ysjeB P sXsWrtZ8dx	OJjzp6C K FXueIN8PcJg3KFe7 juKiaqTIVoTAEMdtiV0sA1Ew4ZPAwV c MXB0VXrEErDH yV w6Ryf 7l v pX
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	460001	460001


- Continue synchronizing the objects in the transaction until all objects in the transaction are synchronized. (Repeat steps 3 and 4.) When all objects in the transaction are synchronized, **Mark Resolved** appears at the bottom of the page, as Figure 2-8 shows.

Figure 2-8: When All Conflicts Are Resolved, the Mark Resolved Button Appears

DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566510001
DeviceChangeLog 470001		
DB Field	C34	C33
CHANGE_SUMMARY	rWC1ysjeB P sXsWrtZ8dxZY10QvHR3KaQxGSWcG0IPqz 0CCgE7I31tgKA5rAftyPrZX LJChwR WV85QxGj6k W zL eqic	rWC1ysjeB P sXsWrtZ8dxZY10QvHR3KaQxGSWcG0IPqz 0CCgE7I31tgKA5rAftyPrZX LJChwR WV85QxGj6k W zL eqic
CONFLICTING	0	0
DVC_CHANGE_LOG_ID	470001	470001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566510001
<input type="button" value="Mark Resolved"/>		

- Click **Mark Resolved**. The Multimaster Tools: Mark Conflict Resolved page appears, as Figure 2-9 shows. The page displays the results of marking a transaction resolved.

Figure 2-9: Multimaster Tools Mark Conflict Resolved Page

Multimaster Tools: Mark Conflict Resolved 566530001 		
Return to Conflict Resolution		
All conflicts successfully marked resolved.		
Facility	Conflict ID	Status
C34	6140002	OK
C33	566530001	OK

After it is marked resolved, the transaction disappears from the State and Conflicts views after Opware SAS refreshes the data in the Multimaster Tools.

- Click the link to return to the Conflict view.

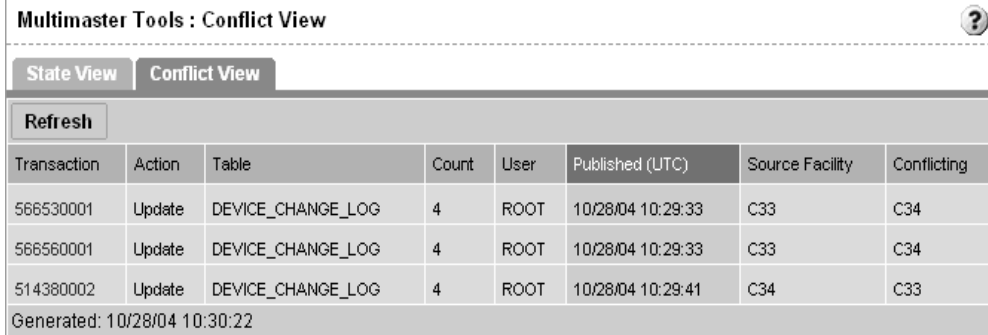
Resolving a Conflict by Transaction

Perform the following steps if you know that synchronizing all objects from one facility will resolve the conflict:

- From the navigation panel, click Administration ► Multimaster Tools. The Multimaster Tools: State View page appears, showing a summary of all transactions and, if they exist, all conflicts.

- 2** To resolve a conflict, select the Conflict View tab. The Multimaster Tools: Conflict View page appears, as shown in Figure 2-10.

Figure 2-10: Transaction Differences Page That Lists all Transactions In Conflict



Transaction	Action	Table	Count	User	Published (UTC)	Source Facility	Conflicting
566530001	Update	DEVICE_CHANGE_LOG	4	ROOT	10/28/04 10:29:33	C33	C34
566560001	Update	DEVICE_CHANGE_LOG	4	ROOT	10/28/04 10:29:33	C33	C34
514380002	Update	DEVICE_CHANGE_LOG	4	ROOT	10/28/04 10:29:41	C34	C33

Generated: 10/28/04 10:30:22

The page lists each transaction by ID number (clickable link), the actions that caused the conflict, the database objects affected by the conflict, the user responsible for the conflict (listed by the IP of the OCC where the user made the change), when the offending action occurred, the source facility that originated the transaction, and the facilities where the transaction conflicted.

- Click the link of the transaction you want to resolve. You now see the Multimaster Tools: Transaction Differences page, as shown in Figure 2-11.

Figure 2-11: Transaction Differences Page for Multimaster Tools Showing Conflicts Between Facilities

Multimaster Tools: Transaction Differences | 566560001 from Source Facility C33 ?

[Return to Conflict View](#)

Synchronize all objects from C34

DeviceChangeLog 480001

DB Field	C34	C33
CHANGE_SUMMARY	q79abGGQXr pMqtRL JRA9S9AhdQo4AwBuG fQQFK16LJQ6E FJqpe89 Pdsf IYgCDBZbDB fYopa eM9Jw wQODc6 s KkJracIV U7wxdx 22 XBz0R bbYYN LhbkhwwlbjZHPsyM4yqWwRQWIZMIE09GLvqTZQoaVctOg5w gj XJ8dn D7o a	
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	480001	480001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566540001	566600001
	<input type="button" value="Synchronize From"/>	<input type="button" value="Synchronize From"/>

DeviceChangeLog 490001

DB Field	C34	C33
CHANGE_SUMMARY	vm9L8F z fFz8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu LxKq Q5tk8E1NEn iY97Nk GsrRVzlrC9vtIG7O N	jnlpeQuuM YPFNFH2cT 0wspWwYpZDGL9doTSvm9L8F z fFz8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	490001	490001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566540001	566600001
	<input type="button" value="Synchronize From"/>	<input type="button" value="Synchronize From"/>

DeviceChangeLog 500001

- 4 From the Synchronize all objects from drop-down list at the top of the page, select the facility to use as the correct source of data, as Figure 2-12 shows.

Figure 2-12: by Transaction

Multimaster Tools: Transaction Differences | 566560001 from Source Facility C33

[Return to Conflict View](#)

Synchronize all objects from

See “Best Practices for Resolving Database Conflicts” on page 74 in this chapter for more information

- 5 Click **Update** beside the drop-down list. The Multimaster Tools: Transaction Synchronization Results page appears, as shown in Figure 2-13.

Figure 2-13: Transaction Synchronization Results For All Objects in Transaction

Multimaster Tools: Transaction Synchronization Results | 566560001 ?

[Return to Conflict Resolution](#)

Transaction successfully synchronized.

Table	Facility	Action
DeviceChangeLog 480001	C34	Unlock
	C33	Update
DeviceChangeLog 490001	C34	Unlock
	C33	Update
DeviceChangeLog 500001	C34	Unlock
	C33	Update
DeviceChangeLog 510001	C34	Unlock
	C33	Update

This page shows the results of the synchronization and prompts you to mark the conflicts resolved.

- 6 Click **Mark Resolved**. The Multimaster Tools: Mark Conflict Resolved page appears. The page displays the results of marking a transaction resolved.
- 7 Click the link to return to the Conflict view. After it is marked resolved, the transaction disappears from the State and Conflicts views after Opsware SAS refreshes the data in the Multimaster Tools.

Network Administration for Multimaster

Opware SAS does *not* require that a multimaster configuration meet specific guidelines on network uptime. A multimaster configuration functions acceptably in a production environment that experiences temporary inter-facility network outages.

However, as the duration of a network outage increases, the probability of multimaster conflicts increases. Extended network outages between facilities can cause the following problems:

- Multimaster messages fail to propagate between facilities.
- The Multimaster Tools stop functioning.
- Opware Command Centers cannot contact the multimaster central Data Access Engine.

Production experience for multimaster configurations supports the performance data that Table 2-1 shows.

Table 2-1: Performance Data for Multimaster Configurations

# FACILITIES	DURATION NETWORK OUTAGE	# MULTIMASTER CONFLICTS *
8 facilities (Opware core installed in each facility)	12 hour outage (1 facility loses network connectivity to the other facilities)	12 to 24 conflicts (average number generated)
* The propensity of users to manage servers in the disconnected facility with Opware Command Centers in other facilities increases the number of conflicts.		

Network connectivity issues include TIBCO or multicast routing problems.

Multimaster Alert Emails

When multimaster conflicts occur or multimaster components experience problems, Opware SAS sends an email to the configured multimaster email alias.

This email address is configured when Opware SAS is installed in a facility. For assistance changing this email address, contact your Opware SAS Support Representative or see the *Opware® SAS Configuration Guide*.

The subject line of the alert email specifies:

- The type of error that occurred when a transaction was being applied to a Model Repository database
- The type of error that caused problems with the multimaster operation

Contact your Opsware SAS Support Representative for assistance troubleshooting and resolving Opsware SAS problems that affect the multimaster operation.

See Table 2-2 for error messages.

Table 2-2: Multimaster Error Messages

SUBJECT LINE	TYPE OF ERROR	DETAILS
vault.ApplyTransactionError	Multimaster Transaction Conflict	The local database was not successfully updated with the changes from the other database. Each update must affect only one row and not result in any database errors.
vault.configValueMissing	Opsware SAS Problem	No value was specified for a given configuration parameter. Log into the Opsware Command Center and provide the value for this configuration parameter. Contact your Opsware SAS Support Representative for assistance setting Opsware SAS configuration values.
vault.DatabaseError	Multimaster Transaction Conflict	An error occurred while querying the database for updates to send to other databases or while applying updates from other databases. Restart the Model Repository Multimaster Component.

Table 2-2: Multimaster Error Messages

SUBJECT LINE	TYPE OF ERROR	DETAILS
vault.InitializationError	Opware SAS Problem	<p>An error occurred when the Model Repository Multimaster Component process started. The application returned the message specified. The thread that encountered the error stopped running. This error occurs when running Opware SAS in multimaster mode.</p> <p>Resolve the error condition. Restart the Model Repository Multimaster Component.</p>
vault.ParserError	Multimaster Transaction Conflict	<p>An error occurred when parsing the XML representation of the transaction. The application returned the message specified. This error occurs when running Opware SAS in multimaster mode.</p> <p>Run the Opware Admin Multimaster Tools and verify that the transaction data does not contain special characters that the XML parser might be unable to interpret.</p>

Table 2-2: Multimaster Error Messages

SUBJECT LINE	TYPE OF ERROR	DETAILS
vault.SOAPError	Multimaster Transaction Conflict	<p>An error occurred while using SOAP libraries to marshal or unmarshal transactions into XML. The application returned the message specified. This error occurs when running Opsware SAS in multimaster mode.</p> <p>Run the Opsware Admin Multimaster Tools and verify that the transaction data does not contain special characters that SOAP might be unable to interpret.</p>
vault.TibcoError	Opsware SAS Problem	<p>The TIBCO transport raised an error. The application returned the message specified. The thread that encountered the error stopped running. This error occurs when running Opsware SAS in multimaster mode.</p> <p>Resolve the TIBCO transport error. See the TIBCO User's Guide for information. Restart the Model Repository Multimaster Component.</p>
vault.UnknownError	Opsware SAS Problem	<p>The Model Repository Multimaster Component process encountered an unknown error. Contact technical support and provide the database name and Opsware SAS component's log file.</p>

Chapter 3: Opsware Satellite Administration

IN THIS CHAPTER

This section discusses the following topics:

- Overview of the Opsware Satellite
- Satellite Information and Access
- Software Repository Cache Management
- Creation of Manual Updates

Overview of the Opsware Satellite

With an Opsware Satellite, a full Opsware core is not installed in a remote facility. Instead, an Opsware Gateway and Software Repository Cache are installed. An Opsware Gateway provides network connection and bandwidth management to a Satellite. A Satellite can contain multiple Gateways. The Software Repository Cache contains local copies of software packages to be installed on managed servers in the Satellite. Optionally, a Satellite can contain the OS Provisioning Boot Server and Media Server components.

A Satellite must be linked to at least one core, which may be either standalone or multimaster. Multiple Satellites can be linked to a single core.

For information about how to install and configure a Satellite, see the *Opsware® SAS Deployment and Installation Guide*.

In Figure 3-1, a Satellite is linked to a standalone core via the Gateway and in Figure 3-2, two Satellites are linked to an Opsware core via the Gateway.

Figure 3-1: A Standalone Core with a Single Satellite

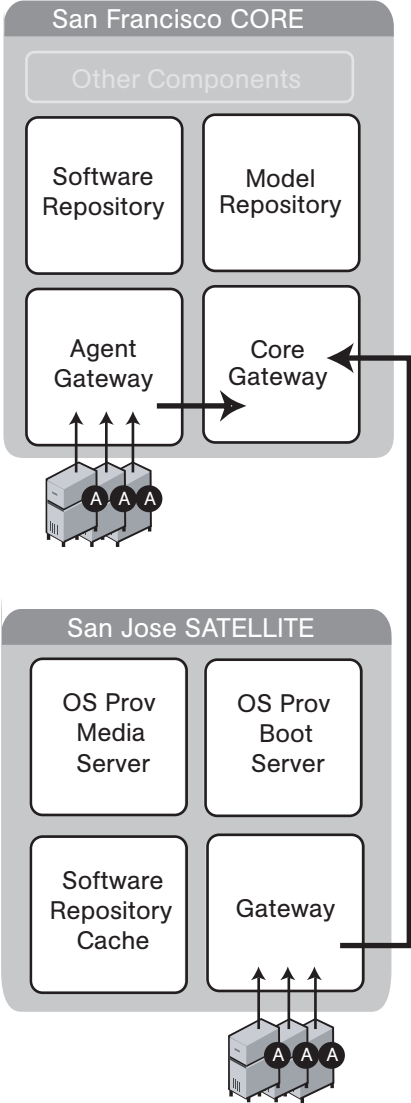
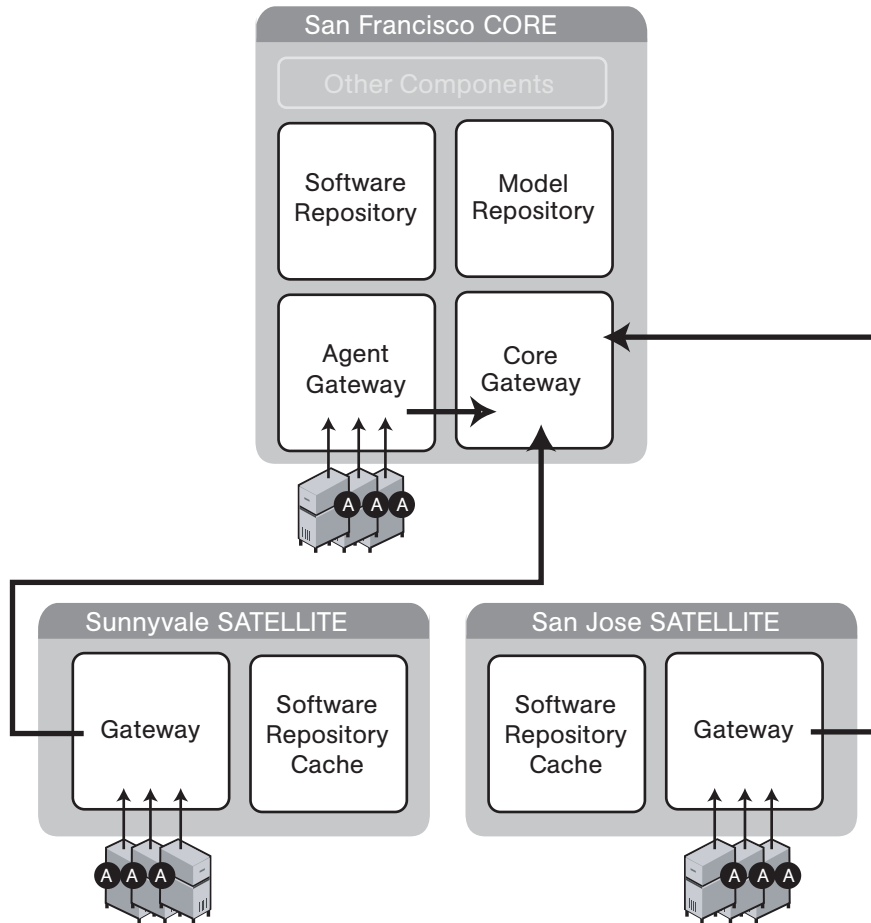


Figure 3-2: Standalone Core with Multiple Satellites



Opware Gateway

Connectivity with an Opware core is achieved through an Opware Gateway that resides in the same IP address space as the servers that it manages. This Opware Gateway maintains a connection to the Opware Gateway in the core, either directly or through a network of Gateways. All traffic between the servers in the Satellite and the core that manages them is routed through Opware Gateways.

Facilities and Realms

To support Opware Agents in overlapping IP address spaces, an Opware core supports realms.

One or more Opware Gateways service the managed servers contained within an Opware realm. In Opware SAS, a realm is a routable IP address space, which is serviced by one or more Gateways. All managed servers that connect to an Opware core via a Gateway are identified as being in that Gateway's realm.

A facility is a collection of servers that reside in a single physical location. A facility can be all or part of a data center, server room, or computer lab. A facility can contain multiple realms to support managed servers with overlapping IP address spaces. Each IP address space requires a separate realm. Typically, each physical building is modeled as a facility that has as many realms as needed.

Satellite Information and Access

This section discusses the following topics:

- Permissions Required for Managing Satellites
- Viewing Facilities
- Viewing the Realm of a Managed Server
- Viewing Gateway Information

Permissions Required for Managing Satellites

To access the Manage Gateway feature, you must have the Manage Gateway permission. By default, this permission is included in the Opware System Administrators group. To view facility information, you must have Read (or Read & Write) permission for the specific facility. See the *Opware® SAS Configuration Guide* for information about user groups and Opware permissions.

Viewing Facilities

The Facilities page in the Opsware Command Center lists the core and Satellite facilities. In particular, the Facilities page displays Unreachable Facilities, as shown in Figure 3-3.

Figure 3-3: Facilities Channel

Facilities

New facility

Select a facility:

Facilities

- GREEN
- SAT1 *
- VIOLET
- WHITE

* Indicates satellite facility

Unreachable Facilities

- SAT2
- TEST
- Test

Clicking the link for a facility, and then selecting the Realms tab displays the configured bandwidth of the connections between the realms in that facility, as shown in Figure 3-4.

Figure 3-4: Realms in Facilities

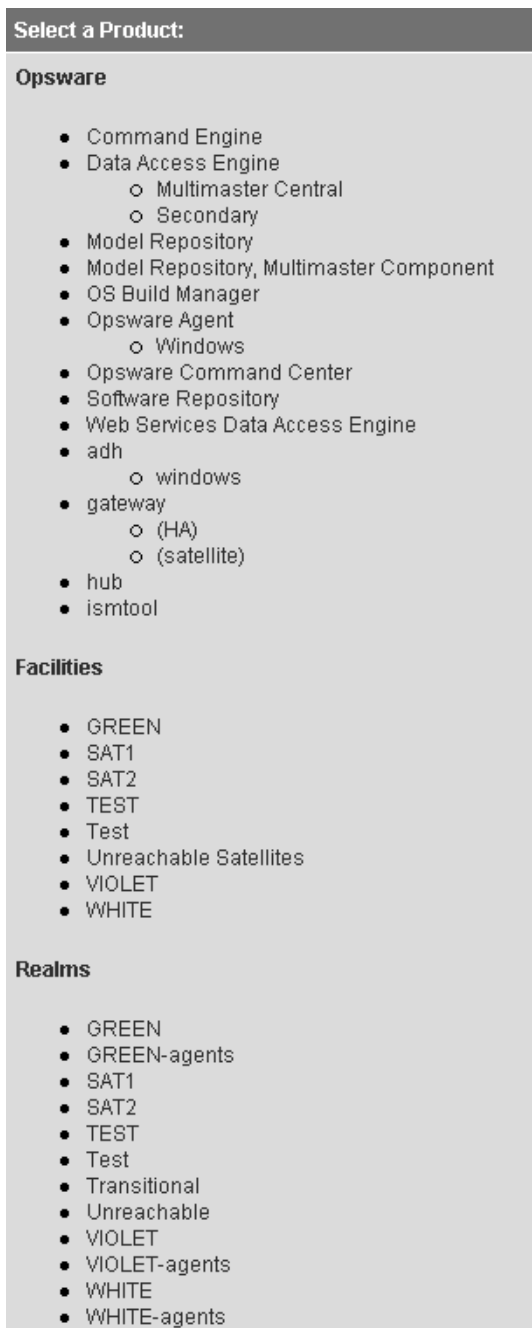
Facilities: Realms for "GREEN"

Return to Facilities

Return to Facilities		
Properties	Custom Attributes	Realms
Name	Bandwidth	
GREEN (Primary)	unlimited	
GREEN-agents	unlimited	

Additionally, you can view the facilities that contain realms by clicking Administration ► System Configuration as shown in Figure 3-5.

Figure 3-5: Satellite Configuration Parameters



Enabling the Display of Realm Information

By default, the Opware Command Center 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 Opware Command Center as a user that belongs the Administrators group and to a group that has the Configure Opware permission.
- 2** From the navigation panel, click Administration ► System Configuration.
- 3** Select the Opware Command Center link.
- 4** In the System Configuration page, for the name `owm.features.Realms.allow`, type the value `true`.
- 5** Click **Save**.

Viewing the Realm of a Managed Server

When installed in a Satellite configuration, Opware SAS can manage servers with overlapping IP addresses. This situation can occur when servers are behind NAT devices or firewalls. Servers with overlapping IP addresses must reside in different realms.

When retrieving a list of servers resulting from a search, you might see multiple servers with the same IP address but in different realms. You might also see multiple servers with the same IP address when you are planning to run a custom extension and you are prompted to select the servers to run the extension on.

The Opsware Command Center displays additional information to make it clear which server contains the server corresponding to the IP address, as shown in Figure 3-6.

Figure 3-6: Server Properties Page Showing the Realm of a Managed Server

Manage Servers: Properties | dhcp-164-5 ?

[Return to Manage Servers](#)

Properties	Network	Membership	Attached Nodes	Installed Packages	Custom Attributes	Config Tracking	History
------------	---------	------------	----------------	--------------------	-------------------	-----------------	---------

MANAGEMENT INFORMATION	
Name:	<input type="text" value="dhcp-164-5"/>
Notes:	<input type="text" value=""/> <input type="button" value="↑"/> <input type="button" value="↓"/>
IP Address:	192.168.164.5
OS Version:	Windows 2000
Customer:	<input type="text" value="Not Assigned"/>
Facility:	SAT1
Realm (Link speed):	SAT1 (56 kbps)
Server Use:	<input type="text" value="Not Specified"/>
Deployment Stage:	<input type="text" value="Not Specified"/>
Config Tracking:	<input type="text" value="Disabled"/>
Console:	(not set)
Opsware Lifecycle:	Managed
Server ID:	510001

Viewing Gateway Information

To access the Manage Gateway feature, click Administration ► Gateway in the Opsware Command Center navigation panel. The Manage Gateway page appears, as Figure 3-7 shows. From the left list, select the Gateway you want to view information for, and then click the link for the page you want to view.

Figure 3-7: Status Page of the Manage Gateway Feature

The screenshot displays the 'Manage Gateway' interface. On the left, a list of gateways includes 'cgw0-C28', which is selected. The main content area shows details for 'Gateway: cgw0-C28' with tabs for Status, Flows, Routing, PathDB, LSDB, Config, History, Ident, Bandwidth, Link Cost, Logging, and Process Control. A 'Page Selection' dropdown is visible. Below this is a table of gateway statistics:

Gateway	Cost	BWLimit Kbits/sec	Send BW Kbits/sec	Recv BW Kbits/sec	Total In Bytes	Total Out Bytes	Payload In Bytes	Payload Out Bytes	Age	Peer
cgw0-C29	1	0	3.21	1.88	382167107	453808297	314905635	396777686	3:5:36:5.46	192.168.196.244:54307
Alice	10	0	1.58	1.23	39021515	56595009	30485950	43693609	3:6:6:9.40	192.168.9.50:41128
agw0-C28	1	0	1.58	0.00	26450755	62224516	25523838	48682818	3:6:5:25.80	127.0.0.1:50991

Below the table are sections for 'Endpoint', 'TunnelMgmt', 'Route', 'MsgProcessor', and 'DataMover Queue Table'. A blue arrow labeled 'Gateway Selection' points to the 'cgw0-C29' entry in the left-hand list.

You use the Manage Gateway feature for the following tasks:

- To obtain debugging and status information about the Gateways and the tunnels between Gateways
- To perform specific tasks for Gateways, such as changing the bandwidth limits or tunnel cost between Gateway instances, restarting Gateway processes, or changing the logging levels for Gateway processes

Viewing Diagnostic and Debugging Information

- 1 From the navigation panel, click Administration ► Gateway. The Manage Gateway page appears.
- 2 From the left list, select the Gateway that you want to view information for. The Status page for that Gateway appears.

The Status page displays the following information for the Gateway:

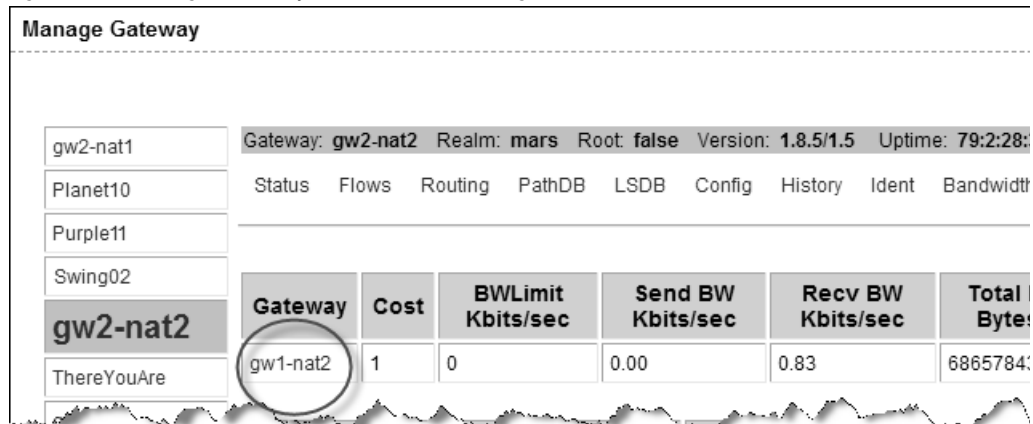
- A table of active tunnels. This table includes tunnel cost, bandwidth constraints, bandwidth estimations, and the age of the tunnels.
- Information about the internal message queues. Each column in the table for a queue displays data in this format:

Number of messages in the queue | The message high-water mark for the queue | Maximum value configured for the queue | The last time the message high-water mark was reached for the queue

You can use the timestamp indicating when the message high-water mark was last reached to troubleshoot Gateway issues. The timestamp is displayed in the format days:hours:minutes: seconds.

- 3 To view the details and statistics for a tunnel between Gateways, click the link for the Gateway that terminates the tunnel, as Figure 3-8 shows.

Figure 3-8: Manage Gateway Feature – Status Page



The page refreshes and displays the tunnel details and statistics.

- 4 To view the following pages containing diagnostic information, click the link for the page in the menu bar.

- **Flows page:** Displays information about all open connections for the selected Gateway.

- **Routing page:** Displays the inter-Gateway routing table. This table shows which tunnel will be used to reach another Gateway in the mesh. The routing table is computed from the data in the path database.

When a tunnel collapses, the route information is retained for 2 minutes by default in the routing table to provide some inertia and stability for the Gateway mesh.

The routing computation automatically updates when the link cost for a connection is changed.

- **Path database (PathDB) page:** Displays the least cost route to all reachable Gateways in the Gateway mesh. The least cost route to all reachable Gateways is determined by using the data in the link state database.
- **Link State database (LSDB) page:** Displays information for the state of all tunnels from the perspective of each Gateway instance. The LSDB contains the data for all tunnels and the bandwidth constraint for each tunnel.
- **Configuration (Config) page:** Displays the properties file for the Gateway you are viewing information for. The page includes the path to the properties file on the server running the Opsware Gateway component.

Below the properties values, the page contains crypto file information and the mesh properties database.

Above the properties values, the Properties Cache field appears. When you change the bandwidth or link cost for a connection between Gateways, the updated value appears in this field if the update was successful.

- **History:** Displays historical information about the inbound (ingress) and outbound (egress) connections between hosts using the Gateway mesh. For example, when host A in realm A connected to host B in realm B.

Finding the Source IP Address and Realm for a Connection

The Ident page provides an interface to the real-time connection identification database. If necessary, contact Opsware Support for additional information about how to run this tool.

- 1** From the navigation panel, click Administration ► Gateway. The Manage Gateway page appears.
- 2** From the top bar (the page selector), click Ident. The page refreshes with an interface to the real-time connection identification database.

3 In the text field, enter the protocol and source port for an active connection (for example, TCP:25679).

4 Click **Lookup**.

The page refreshes with the client realm and client IP address – where the connection came from.

Changing the Bandwidth Usage or Link Cost Between Gateways

1 From the navigation panel, click Administration ► Gateway. The Manage Gateway page appears.

2 To set a bandwidth limit for a connection:

1. From the top bar (the page selector), click Bandwidth. The page refreshes with fields to specify the bandwidth for the connection between Gateway instances.
2. Specify two Gateway instance names that are connected by a tunnel.
3. Specify the bandwidth limit you want in kilobits per second (Kbps). Specify zero (0) to remove bandwidth constraints for the connection.
4. Click **Apply**.

3 To set a link cost for a connection:

1. From the top bar (the page selector), click Link Cost. The page refreshes with fields to specify the link cost for the connection between Gateway instances.
2. Specify two Gateway instance names that are connected by a tunnel.
3. Specify the cost you want in the Cost field.
4. Click **Apply**.

Viewing the Gateway Log or Change the Log Level



Changing the logging level to LOG_DEBUG or LOG_TRACE greatly increases the log output of the Gateway and can impact the performance of the Gateway.

1 From the navigation panel, click Administration ► Gateway. The Manage Gateway page appears.

2 From the top bar (the page selector), click Logging. The page refreshes with the tail of the Gateway log file.

- 3** To change the logging level, select an option: LOG_INFO, LOG_DEBUG, or LOG_TRACE.
- 4** Click **Submit**.

Restarting or Stopping a Gateway Process

- 1** From the navigation panel, click Administration ► Gateway. The Manage Gateway page appears.
- 2** From the top bar (the page selector), click Process Control. The page refreshes.
- 3** To restart the Gateway process, click **Restart**.
- 4** To stop the Opsware Gateway watchdog and the Opsware Gateway, click **Shutdown**.



Stopping a Gateway process can cause problems for an Opsware core. For example, if you stop a core Gateway process, you will stop all multimaster traffic to that Opsware core. Additionally, the Manage Gateway UI is unavailable after stopping the process.



To restart the Gateway after stopping it from the Manage Gateway page, you must log onto the server running the Opsware Gateway component and manually restart the process.

Software Repository Cache Management

The largest amount of traffic in an Opsware core is between the Software Repository and the Opsware Agent (during software or patch installation) and between a server being provisioned and the media server servicing the installation.

When a Satellite is connected by a low-bandwidth network link, during software installation on servers Opsware SAS performance in the Satellite will be poor unless special steps are taken, for example, installing a 1GB software package onto a server in a Satellite connected by a 56 kbps link will take a long time.

By placing a local copy of the Software Repository and OS installation media local to the Satellite in a Software Repository Cache, bandwidth utilization can be optimized. In a Satellite, the Software Repository Cache contains copies of files that are local to the Satellite.

The Software Repository Cache stores files from the Software Repository in an Opsware core or from another Software Repository Cache, and supplies the cached files to Opsware Agents on managed servers. The Opsware Satellite supports multiple Software Repository Cache per realm.

Availability of Packages on the Software Repository Cache

All content, such as patches, software updates, and so on, might not be available locally at all Satellites. Opsware SAS indicates whether a package is available locally or whether the Satellite needs to obtain an update from the Software Repository in the Opsware core. The Opsware Command Center does not proactively warn you that software installation will fail because the package is unavailable locally and caching constraints do not allow On-demand Updates.

Instead, when Opsware SAS is attempting to reconcile the software onto a managed server, the Opsware Command Center generates an Opsware error and displays a complete list of missing packages to help you identify the packages that need to be staged.

To view whether a package is available in a specific Satellite (realm), perform the following steps:

- 1** From the Opsware Command Center left navigation bar, click Packages under Software. The list of packages appears.
- 2** Click the link of the package to check its availability. The properties for the package appears.
- 3** Select the Availability tab.

As Figure 3-9 illustrates, the Availability tab for software packages displays the realms in which that package is available and the bandwidth between the realm and the Opsware core.

Figure 3-9: Package Availability

Package: View Availability | 106541-23

Return to Browse Packages

Properties	Nodes	Availability
This package is available in the following Realms:		
Name	Bandwidth	
GREEN	unlimited	
VIOLET	unlimited	
WHITE	unlimited	



The Opsware Command Center does not provide a User Interface to push packages to Satellites. To push packages to a Satellite, the command-line tool `stage_pkg_in_realm` may be used. This tool is found on the wordbot in `/cust/usr/blackshadow/mm_wordbot/util`. The Software Repository Cache allows a client to request that it obtain a file, but that it not actually send the file to the client. If the file is not already cached, the Software Repository Cache will obtain it from the parent Software Repository Cache if the caching policy allows it. To use this feature, the client includes the argument `checkonly=1` in the URL request for the file.

Ways to Distribute Packages to Satellites

To update files in a Satellite, the Software Repository Cache in that facility can be configured to update cached copies of files as requests are received (On-demand Updates) or to update the cached copy of a file manually (Manual Updates):

- **On-demand Update:** The local Software Repository Cache obtains current files when needed from the Software Repository in the Opsware core.

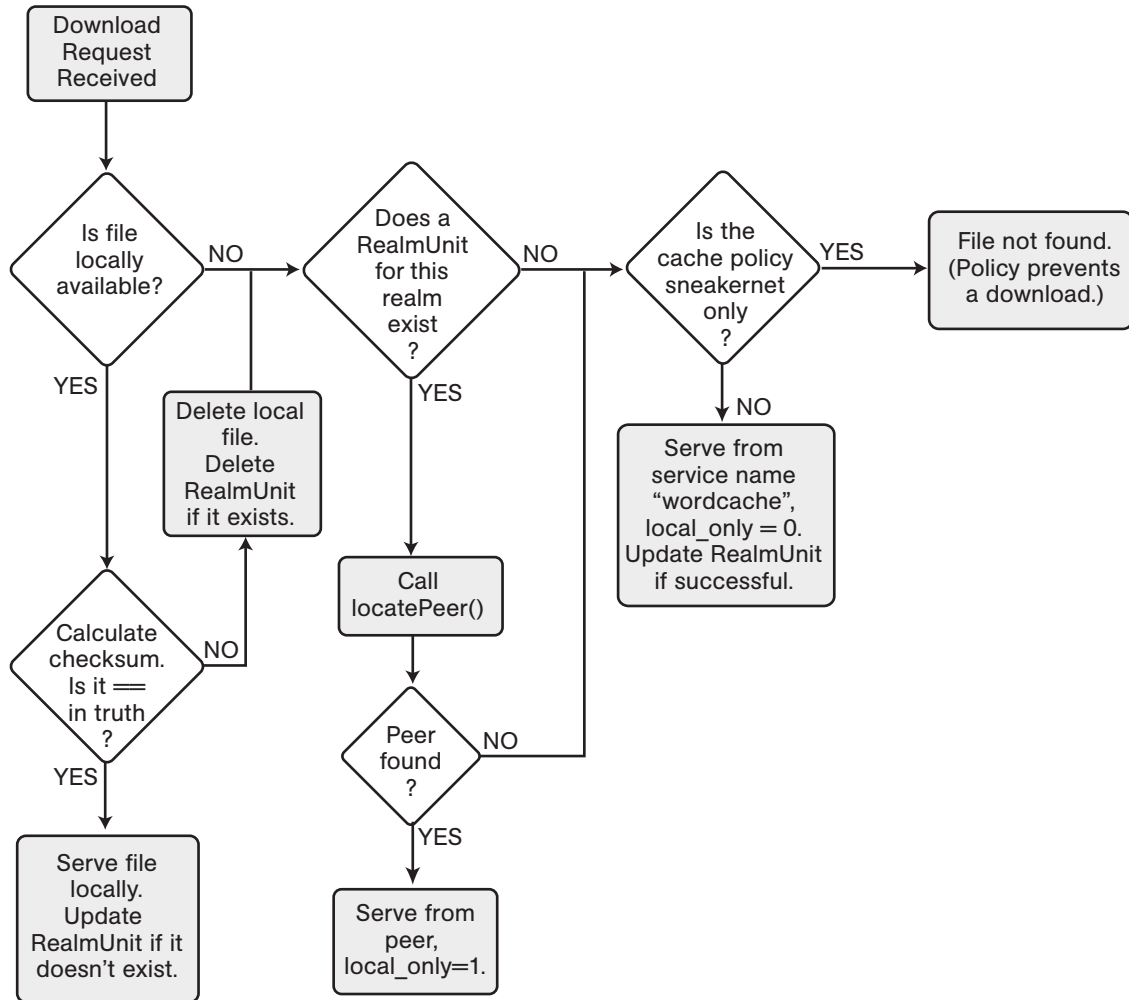
- **Manual Update:** Software packages are staged to a Satellite's Software Repository Cache in advance of package installation so that performance will be about the same as if the managed server was in the same data center as the core.

It is always possible to stage a file on a Software Repository Cache regardless of the caching policy. See "Staging Files to a Software Repository Cache" on page 116 in this chapter for more information.

If the file is already present on the local Software Repository Cache and is current, no action will be taken. If the file is not present locally or it is not current, the Software Repository Cache will attempt to download the file in the background from the upstream Software Repository Cache or Software Repository. If the caching policy for the realm of the Software Repository Cache is on-demand, the download will be successful. If the caching policy is Manual Update, the Software Repository Cache will raise a `wordbot.unableToCacheFile` exception.

The flowchart in Figure 3-10 illustrates the logic that the Software Repository Cache uses to update packages in a Satellite.

Figure 3-10: Software Repository Cache Update Logic



Setting the Update Policy


You can specify the Software Repository Cache update policy for specific facilities by performing the following steps:

- 1** From the Opsware Command Center navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 2** Click the link of the realm for which you want to set the Software Repository Cache update policy. The configuration values for that facility appear.
- 3** For the parameter named `word.caching_policy`, set the caching policy value by selecting the Use default value option or the Use value option and typing SNEAKERNET, as shown in Figure 3-11. In the Opsware Command Center, On-demand Update is referred to as Just-in-time (JIT) and Manual Update is referred to as Sneakernet.

Figure 3-11: Software Repository Cache Configuration Parameters

System Configuration: Set Configuration parameters ?

Return to System Configuration

 These configuration parameters should be changed only under the direction of Opsware, Inc.

Modify configuration parameters for: Realms > SAT1

Name	Value
osprov.stage2_host: null	<input checked="" type="radio"/> Use default value: buildmgr <input type="radio"/> Use value: <input style="width: 150px;" type="text"/> ⌵ ⌴ ...
word.caching_policy: Caching policy for the word. Either JIT or SNEAKERNET.	<input type="radio"/> Use default value: JIT <input checked="" type="radio"/> Use value: <input style="width: 150px;" type="text" value="SNEAKERNET"/> ⌵ ⌴ ...
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

- 4** Click **Save** to apply your configuration change. Since the Software Repository Cache polls for configuration changes every five minutes (by default), it may take up to five minutes for your change to take effect.

On-demand Updates

Each time an Opsware Agent on a managed server in a Satellite requests a package, the local Software Repository Cache checks the currentness of its cached copy of the file. If the cached file is out of date (or missing), the Software Repository Cache obtains an updated copy of the file from the upstream Software Repository Cache or from the Software Repository in the core and sends it to the Opsware Agent.

When configured for On-demand Updates, the Software Repository Cache requests the checksum of each requested file from the Opsware Model Repository.



For security purposes, Opsware SAS caches the checksums about the currentness of a file for a configurable period of time only.

If the checksum is the same as the locally-stored file, the Software Repository Cache serves the file to the requester. If the checksum does not match or the local file is not present, the Software Repository Cache requests a copy of the file. The Opsware Gateway routes the request to the upstream Software Repository Cache in the Gateway hierarchy or to the Software Repository if no upstream Software Repository Cache exists.

If network connectivity is lost while the Software Repository Cache is downloading a file from an upstream Software Repository Cache or from the Software Repository in the core, the next time an Opsware Agent requests the same file, the Software Repository Cache will resume the file download from the point it stopped.

Manual Updates

In Satellites that are behind low-bandwidth network links, the Manual method for updating a Software Repository Cache can be used to pre-populate a cache at installation time or to refresh a cache. The Software Repository Cache is populated by an out-of-band method, such as by cutting CDs of the required packages and shipping them to the Satellite.

When configured for Manual Updates, a Software Repository Cache does not communicate with upstream Software Repository Cache or the Software Repository in the core unless requested. It treats its cache as authoritative.

Emergency updates can still be manually pushed over the network to Satellites even if the caching policy is Manual only Update. You do not need to reconfigure the Software Repository Cache's caching policy to push emergency updates to a Software Repository Cache. For example, an emergency patch can be staged to a Satellite and applied without waiting for a shipment of CDs to arrive.

The Opsware Command Center displays a warning when a user stages a package to a Software Repository Cache that is configured for Manual Update.

Additionally, a Manual Update can be applied to any Software Repository Cache regardless of its update policy.

When applying manual updates in a Satellite with multiple Software Repository Caches, you must apply the update to each Software Repository Cache in the Satellite. Otherwise, when performing operations that retrieve files from the Cache (for example, when installing software on a server in the affected Satellite), you may get the `wordbot.unableToCache file` error.

Hierarchical Software Repository Caches

When Opsware SAS contains hierarchal realms, each realm can contain a local Software Repository Cache.

When an Opsware Agent requests an unavailable file from its local Software Repository Cache, the Software Repository Cache checks its configuration to see if it is allowed to perform an On-demand Update. If configured for updates, the request is passed up the topology chain only until the requested file is found or until a Software Repository Cache is configured for Manual Updates.

If the file is unavailable because of the caching policy, you can stage the file to the local Software Repository Cache. Because of this behavior, Manual Updates need only be applied to the top-level Software Repository Cache within a Manual Update only zone.

Cache Size Management

If you apply a Manual Update to a Software Repository Cache configured for Manual only updates, the Software Repository Cache will remove files that have not been recently accessed when the cache size limit is exceeded.

When the Software Repository Cache exceeds the cache size limit, the least-recently accessed packages are deleted first, regardless of whether they are current or not.

The Software Repository Cache removes the files the next time it cleans up its cache. By default, the cache is cleaned up every 12 hours. Packages are deleted so that the available disk space goes below the low water mark.



Opware recommends that customers have enough disk space to store all necessary packages for the Software Repository Cache to ensure that the Software Repository Cache does not exceed the cache size limit.

Creation of Manual Updates

To create a Manual Update, you can use the Opware DCML Exchange Tool (DET) to copy existing packages from an Opware core. You can then save the exported file to CD or DVD to apply later to a Satellite Software Repository Cache.

This section discusses the following topics:

- Creating a Manual Update Using the DCML Exchange Tool (DET)
- Applying a Manual Update to a Software Repository Cache
- Staging Files to a Software Repository Cache
- Microsoft Utility Uploads and Manual Updates

Creating a Manual Update Using the DCML Exchange Tool (DET)

You perform this procedure by using the DCML Exchange Tool (DET). Using the Opware DET, you export the packages you want for the Manual Update and export the packages associated with selected software tree nodes.

See the *Opware[®] SAS DCML Exchange Tool (DET) Reference Guide* for more information about the DET.

To create a Manual Update perform the following steps:

- 1** On the server where you installed the DET component, enter the following command to create the following directory:

```
mkdir /var/tmp/sneakernet
```

- 2** From the server running the Opware Command Center component in the Opware core, copy the following files from the `/var/lc/crypto/owm` directory:

opsware-ca.crt

spog.pkcs.8

to the following directory:

/usr/cbt/crypto

This is the directory where you installed the DET.

- 3** Create the following file `/usr/cbt/conf/cbt.conf` so that it contains this content:

```
twist.host=<twist's hostname>
twist.port=1032
twist.protocol=t3s
twist.username=buildmgr
twist.password=buildmgr
twist.certPaths=/usr/cbt/crypto/opsware-ca.crt
spike.username=<your username>
spike.password=<your password>
spike.host=<way's hostname>
way.host=<way's hostname>
spin.host=<spin's hostname>
word.host=<word's hostname>
ssl.keyPairs=/usr/cbt/crypto/spog.pkcs8
ssl.trustCerts=/usr/cbt/crypto/opsware-ca.crt
```

- 4** Create the following DCML Exchange Tool filter file `/usr/cbt/filters/myfilter.rdf` that contains this content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
<!ENTITY filter "http://www.opsware.com/ns/cbt/0.1/filter#">
]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://www.opsware.com/ns/cbt/0.1/filter#">
<ApplicationFilter rdf:ID="a1">
<path>/Other Applications</path>
<directive rdf:resource="&filter;Descendants" />
</ApplicationFilter>
</rdf:RDF>
```

In the `<path>` directive of the filter file, replace `/Other Applications` with the path to the node you want to export (all node information about that node, its descendants, and all associated packages will be exported).

This filter will export from the Applications area of the Opsware Command Center. If you want to export packages from some other category of software in the Opsware Command Center, you need to create a different filter. See the *Opsware® SAS DCML Exchange Tool (DET) Reference Guide* for information.

- 5** On the server where you installed the DET component, run the DCML Exchange Tool by entering the following command:

```
/usr/cbt/bin/cbt -e /var/tmp/myexport --config /usr/cbt/conf/cbt.conf --filter /usr/cbt/filters/myfilter.rdf
```

The DCML Exchange Tool places the packages associated with the exported nodes in the following directory:

```
/var/tmp/myexport/blob
```

The packages are named `unitid_nnnnnnnn.pkg`.

- 6** Copy all of the `.pkg` files to a directory on the server running the Software Repository Cache, either over the network or by burning the files to a set of CDs.

Applying a Manual Update to a Software Repository Cache

To apply a Manual Update to a Software Repository Cache, you run an Opsware utility (`import_sneakernet`), which moves or copies the packages you want to update into the right location on the Software Repository Cache and registers them with the Opsware Model Repository in the Opsware core.

To apply a Manual Update to a Software Repository Cache, perform the following steps:

- 1** Log in as root on the server running the Software Repository Cache in the Satellite.
- 2** Mount the CD containing the packages or copy them to a temporary directory.
- 3** Enter the following command to change directories:

```
cd /cust/usr/blackshadow/mm_wordbot/util
```

- 4** Enter the following command to import the contents of the Manual update to the Software repository Cache:

```
./import_sneakernet -d dir
```

where `dir` is the CD mount point or the temporary directory containing the packages.

Staging Files to a Software Repository Cache

The Software Repository Cache allows an Opware Agent on a managed server to override the caching policy in effect for the realm. The ability to override the caching policy of a Software Repository Cache allows you to stage a file to a Manual Update only Satellite in the following types of situations:

- You need to circulate an emergency patch when you do not have time to create a Manual update set and physically visit the facility.
- A necessary patch will be installed during a specified maintenance time period and the time period is not long enough to download the patch and install it on all managed servers.
- The utilization of the network link to the Satellite is known to be low at a particular time of day.

To force package staging, the client includes the argument `override_caching_policy=1` in the URL request for the file.

The Software Repository Cache allows a client to request that it obtain a file, but that it not actually send the file to the client. If the file is not already cached, the Software Repository Cache will obtain it from the parent Software Repository Cache if the caching policy allows it. To use this feature, the client includes the argument `checkonly=1` in the URL request for the file.

Running the Staging Utility

To run the staging utility, perform the following steps:

- 1** On the server running the Software Repository component, verify that the certificate `token.srv` is in your `CRYPTO_PATH`. During installation `token.srv` is copied to `/var/nc/crypto/gateway/token.srv`.
- 2** Log into the server running Opware Software Repository component.
- 3** Enter the following command to change directories:

```
cd /cust/usr/blackshadow/mm_wordbot/util
```
- 4** To stage the files you want, run the utility `stage_pkg_in_realm` which has the following syntax:

```
./stage_pkg_in_realm [-h | --help] [-d | --debug]
[--user <USER>] --pkgid <ID> --realm <REALM> [--gw
<IP:PORT>] [--spinurl <URL>] [--wayurl <URL>] [--word
<IP:PORT>]
```

Example: Command to Run the Staging Utility

```
./stage_pkg_in_realm --user admin --pkgid 80002 --realm luna --
gw 192.168.164.131:3001
```

Password for admin: *<password>*

Package /packages/opsware/Linux/3ES/miniagent is now being staged in realm luna

Microsoft Utility Uploads and Manual Updates

When you upload new Microsoft utilities, including the Microsoft Patch Database (*mssecure.cab*), the Microsoft Baseline Security Analyzer (*mbsaccli.exe*), or the Windows *chain.exe* utility to the Software Repository, you should immediately stage those files to all realms where the Software Repository Cache is configured for Manual only Updates.

If you do not stage these files to the remote realms, Opsware Agents running on Windows servers in those realms will be unable to download new versions of the utilities and will be unable to register their software packages. It is not necessary to stage packages to realms where the Software Repository Cache is configured for On-demand Updates.

The Software Repository Cache allows a client to request that it obtain a file, but that it not actually send the file to the client. If the file is not already cached, the Software Repository Cache will obtain it from the parent Software Repository Cache if the caching policy allows it. To use this feature, the client includes the argument *checkonly=1* in the URL request for the file. See “Running the Staging Utility” on page 116 in this chapter for information about how to stage files.

Chapter 4: Opware SAS Maintenance

IN THIS CHAPTER

This section discusses the following topics:

- Possible Opware SAS Problems
- Opware SAS Diagnosis
- Logs for Opware Components
- Global Shell Audit Logs
- Overview of Restarting Opware Components
- Opware Software
- Mass Deletion of Backup Files
- Designations for Multiple Data Access Engines
- Web Services Data Access Engine Configuration File
- Adding Locales to the Opware Command Center Component
- Automatically Importing Windows Patches

Possible Opware SAS Problems

This section provides information about possible Opware SAS problems and contains the following topics:

- Possible Opware SAS Problems
- Opware Component Troubleshooting
- Contacting Opware Support

While maintaining Opware SAS, you might encounter the following types of problems:

- Operational problems: processes failing or becoming unresponsive (Data Access Engine, Command Engine, Software Repository)
- Failure of an Opware component, which causes other components to fail

The following examples describe the effects of some component failures:

- If the Data Access Engine fails, the Opsware Command Center, the Command Engine, and the Software Repository components will fail.
- If the Software Repository fails to contact the Data Access Engine, downloads from the Software Repository are impossible.
- If the Model Repository fails, the Data Access Engine fails.
- The Software Repository fails to contact the Data Access Engine without either a functioning DNS, or a properly-configured `/etc/hosts` file.
- Unreachable servers existing in the managed environment.



Many problems with the Code Deployment & Rollback (CDR) feature are caused by errors with the CDR configuration and setup. See the *Opsware® SAS Configuration Guide* for information about CDR configuration.

Opsware Component Troubleshooting

The following mechanisms for troubleshooting Opsware SAS are available:

- Running Opsware SAS Diagnosis tool (a tool for debugging common problems with Opsware components). See “Opsware SAS Diagnosis” on page 121 in this chapter for more information.
- Reviewing error logs for Opsware components. See “Logs for Opsware Components” on page 127 in this chapter for more information.
- Contacting Opsware Support.

Contacting Opsware Support

When you contact Opsware Support have the following information available to help you with your support call:

- Be at your computer and have network access to the servers running the Opsware core.
- Have your Opsware guides available.
- Write down the steps followed prior to the problem occurring.

- Write down the exact text of the error that appears on your screen or print the page on which the error appears.
- Be able to describe the problem in detail.

Contact Opsware Technical Support:

Phone: +1 877 677-9273 (1-877-Opsware), in the United States

International Phone: 1 408-212-5300

Email: support@opsware.com

Opsware SAS Diagnosis

This section provides information about how to diagnose Opsware SAS problems and contains the following topics:

- Opsware Component Troubleshooting
- System Diagnosis Testing Process
- System Diagnosis Test Components
- Data Access Engine Tests
- Software Repository Tests
- Web Services Data Access Tests
- Command Engine Tests
- Model Repository Multimaster Component Tests
- Running a System Diagnosis of Opsware Components

Opsware SAS Diagnosis Tool Functionality

By using the System Diagnosis tool, you can check the functionality of the Opsware components and the ability of servers running in the managed environment to interact with the Opsware core.

You can troubleshoot most of the errors that occur within the Opsware core by running the Opsware SAS Diagnosis tool.

System Diagnosis Testing Process

The System Diagnosis tool tests the Opsware components first, and then, optionally, tests the servers that you specify, which are running in the managed environment.

The System Diagnosis tool performs intensive tests of the Opsware components, which check the functionality of the Opsware components:

- **Stand-Alone Tests:** The first suite of tests, which tests as much of the functionality of that component as possible without the use of other Opsware components. The Stand-Alone Tests are run to verify a base level of functionality and the component's ability to respond to an XML-RPC call.
- **Comprehensive Tests:** The second suite of tests, which tests the full functionality of each component.

On completion of the Comprehensive Tests, the System Diagnosis tool displays the success of each test, the results, and error information for the tests that failed.

The components are not tested in a specific order; however, the tests generally occur in this order:

- Opsware Agent Stand-Alone Tests
- Opsware Agent Comprehensive Tests
- Component Stand-Alone Tests
- Component Comprehensive Tests

System Diagnosis Test Components

The tests for the components simulate all the functionality that each component represents. In addition to errors, the tests verify that each component is functioning within certain conditions (for example, whether database connections are near maximum on the Data Access Engine).

The System Diagnosis tool tests the following components:

- Data Access Engine
- Software Repository
- Web Services Data Access Engine
- Command Engine
- Opsware Agents on Opsware core servers
- Model Repository Multimaster Component



The System Diagnosis tool does not test the Build Manager.



When using the System Diagnosis function in an environment with multiple facilities, System Diagnosis can only be run on one facility at a time.

Data Access Engine Tests

The following section describes two types of Data Access Engine diagnostic tests: Stand-Alone and comprehensive.

Stand-Alone Tests

- Check for the current Data Access Engine version.
- Check for the current Model Repository database version.
- Obtain a Device object.
- Obtain a MegaDevice object.
- Verifies advanced query functioning.
- Verify a Device object.
- Obtain the list of facilities.
- Obtain the names of the Data Access Engine cronbot jobs.
- Check whether the usage of database connections is below the acceptable level.
- Check whether any database connection has been open more than 600 seconds.
- Check whether the Data Access Engine and Model Repository are in the same facility.
- Verify that all Model Repository garbage-collectors are running when the Model Repository is running in multimaster mode.
- If the Data Access Engine is configured as the central multimaster Data Access Engine:
 - Check whether multimaster transactions are being published.
 - Check whether multimaster transactions are showing up at remote facilities.
 - Check for multimaster transaction conflicts.

Comprehensive Tests

- Test connectivity to the Model Repository on the configured port.

- Test connectivity to the Command Engine on the configured port.
- Test connectivity to the Software Repository on the configured port.

Software Repository Tests

The following section describes two types of Software Repository diagnostic tests: stand alone and comprehensive.

Stand-Alone Tests

None.

Comprehensive Tests

- Test whether a file that is not a package can be uploaded to the Software Repository process that serves encrypted files. This test verifies whether the file is present in the Software Repository file system and that the file size matches the source.
- Verify that a file can be downloaded from the Software Repository.
- Verify whether the Software Repository process that serves unencrypted files is running and serving files.
- Try to download a file without encryption.
- Verify that a package can be uploaded to the Software Repository and that the package is registered with the Model Repository.
- Verify that a package can be deleted from the Software Repository and removed from the Model Repository.

Web Services Data Access Tests

The following section describes two types of Web Services Data Access diagnostic tests: stand-alone and comprehensive.

Stand-Alone Tests

- Connect to the Web Services Data Access Engine and retrieve its version information.

Comprehensive Tests

- Connect to the Web Services Data Access Engine.
- Read a server record from the Model Repository and thereby check connectivity to the Model Repository.

Command Engine Tests

The following section describes two types of Command Engine diagnostic tests: stand alone and comprehensive.

Stand-Alone Tests

- Check the state machine.
- Check session tables.
- Check lock-down status.
- Check for signature failures.
- Check command and service tables.
- Check the facility cache.

Comprehensive Tests

- Check Data Access Engine connectivity.
- Check security signatures.
- Check lock operation.
- Run an internal script.
- Run an external script.

Model Repository Multimaster Component Tests

The following section describes two types of Model Repository Multimaster Component diagnostic tests: stand alone and comprehensive.

Stand-Alone Tests

- Check the ledger state by examining the ledger file.
- Report the total number of messages sent, number of messages still in the ledger file (for example, not confirmed by all listeners), and the sequence number of the last message confirmed by each listener.
- Check the sender health by examining the state of the Outbound Model Repository Multimaster Component.
- Check the receiver health by examining the state of the Inbound Model Repository Multimaster Component.

Comprehensive Tests

None.

Running a System Diagnosis of Opware Components



To access the System Diagnosis tool, you must have Opware administrator privileges. See the *Opware® SAS Configuration Guide* for more information about how to assign user privileges. The Opware Command Center has access to all the Opware Agents running on the Opware component servers.

Perform the following steps to run a system diagnosis of the Opware Components:

- 1** From the navigation panel, click Administration ► System Diagnosis. The System Diagnosis: Begin Diagnosis page appears.
- 2** Select the components that you want to test. By default, all components are selected (the Data Access Engine, the Software Repository, Command Engine, and Web Services Data Access Engine; in multiple core environments, there is also a selection for the Model Repository Multimaster Component). See Figure 4-1.

Figure 4-1: System Diagnosis Page That Shows Opware Components Selected for Testing on the Indicated Facility

System Diagnosis: Perform Diagnosis

Facility: C03 ▼

Specify Diagnosis Options

Select the Opware Components you would like to test in the selected datacenter.

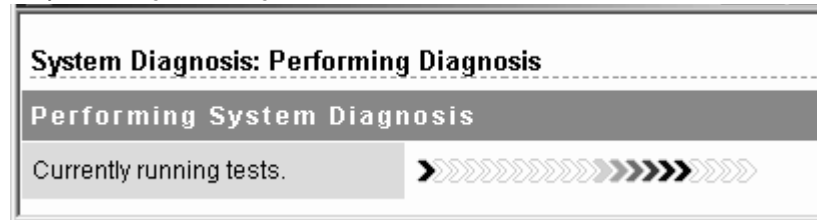
Opware Components:

- Data Access Engine
- Software Repository
- Command Engine
- Model Repository, Multimaster Component
- Web Services Data Access Engine

- 3** Click **Run Diagnosis**.

The System Diagnosis: Performing Diagnosis window appears, which displays a progress bar while the tests are running, as Figure 4-2 shows.

Figure 4-2: System Diagnosis Progress Bar



When all the tests are complete, the window closes and the System Diagnosis: Failed Tests page appears in the main Opware Command Center window. If all tests passed, the System Diagnosis: Successful Tests page appears.

- 4** To review the results of a test, click the linked test name in the Test column. The System Diagnosis: Test Information page appears. If the test contained an error, error information appears at the bottom of the page.

Logs for Opware Components

Opware components record events in log files that are useful for troubleshooting. To view a log file, in a terminal window log into the server running the component and use a command-line utility such as `more`, `grep`, or `vi`.



The log file for a component resides on the server where the component is installed.

By default, the logging debug levels are configured for the highest value (indicating higher priority). The default for the maximum log file size is 10 MB. When the specified maximum file size is reached, additional logs are created. To change the log levels or file sizes, contact your Opware, Inc. support representative for assistance.

Boot Server Logs

The Boot Server does not generate its own logs. The Boot Server uses these services: TFTP with INETD, NFS server, and ISC DHCPD. All of these services log with `syslog`. Consult your vendor documentation for more information on these log files.

Build Manager Logs

These logs are in the following file:

```
/var/lc/buildmgr/buildmgr.log
```

Command Engine Logs

These logs are in the following files:

```
/var/lc/waybot/waybot.err*  
/var/lc/waybot/waybot.log*
```

Data Access Engine Logs

These logs are in the following files:

```
/var/lc/spin/spin.err*  
/var/lc/spin/spin.log*
```



In a core with multiple Data Access Engines, each server running an engine has a set of these log files.

Media Server Logs

These logs are in the following files:

```
/var/opt/OPSWsamba/log.smbd  
/var/opt/OPSWsamba/log.nmbd
```

Solaris and Linux OS provisioning use of vendor-provided services such as NFS. These services typically log through `syslog`. Consult your vendor documentation for more information on these log files.

Model Repository Logs

The Model Repository is an Oracle database. The location logs the database is specific to your installation. For more information, see the Monitoring Oracle Log Files section in the *Opware® SAS Deployment and Installation Guide*.

Model Repository Multimaster Component Logs

These logs are in the following files:

```
/var/lc/vault/err*  
/var/lc/vault/log.*
```


To configure the log file name, log file size, or logging level, in the Opsware Command Center, go to Administration ► System Configuration ► Model Repository Multimaster Component.

Opsware Agents Logs

The Agents create the following log files on managed servers.

Unix:

```
/var/lc/cogbot/cogbot.log*
/var/lc/cogbot/cogbot.err*
```

Windows:

```
%ProgramFiles%Common Files\loudcloud\cogbot\log*
%ProgramFiles%Common Files\loudcloud\cogbot\cogbot.err*
```

Opsware Command Center Logs

The Opsware Command Center does not generate its own logs. The Opsware Command Center uses JBoss server, which writes to the following log files:

```
/var/lc/opswapps/log/server.log*
```

Software Repository Logs

These logs are in the following files:

```
/var/lc/mm_wordbot/wordbot.err*
/var/lc/mm_wordbot/wordbot.log*
/var/lc/mm_wordbot-clear/wordbot-clear.err*
/var/lc/mm_wordbot-clear/wordbot-clear.log*
```

Software Repository Replicator Logs

These logs are in the following files:

```
/var/lc/replicator/replicator.err*
/var/lc/replicator/daemonbot.out
/var/lc/replicator/replicator.log*
```

Software Repository Multimaster Component Logs

These logs are in the following files:

```
/var/lc/mmword/log*
```

Web Services Data Access Engine Logs

The Web Services Data Access Engine contains the following log files:

```
/var/lc/twist/stdout.log*  
/var/lc/twist/twist.log  
/var/lc/twist/access.log  
/var/lc/twist/server.log*  
/var/lc/twist/boot.log  
/var/lc/twist/watchdog.log
```

The `stdout.log` file contains debug output and logging of every exception that the server generates. The file does not conform to a specific format. * indicates the files are `log.1`, `log.2`, `log.3`, and so forth. The number of files and the size of each file can both be configured via `twist.conf`. Additional logs are created when the specified maximum file size is reached. The `stdout.log` is the most recent, and `stdout.log.1` through 5 are progressively older files. The file is also rotated on startup. This file also contains the output of any `System.out.println()`, `System.err.println()` and `e.printStackTrace()` statements.

The `twist.log` file contains JBoss-specific error or informational messages and Weblogic specific messages. These files are rotated on startup.

The `access.log` file contains access information in common log format. These files are rotated when the file reaches 5MB in size.

The `server.log` file contains debug messages generated from the Web Services Data Access Engine. The debug messages are controlled by the log level set at the package or class level in the `twist.conf` file. * indicates the files are `log.1`, `log.2`, `log.3`, and so forth. The number of files and the size of each file can both be configured via `twist.conf`. The `server.log.0` is always the current file, while `server.log.9` is the oldest.

The `boot.log` file contains information on the initial `stdout` and `stderr` messages generated when the Web Services Data Access engine starts. In addition, the `boot.log` file contains the output from `Kill -QUIT` commands.

The `watchdog.log` file records the status of the Web Services Data Access Engine once every minute.

Opsware Gateway Logs

These logs are in the following files:

```
/var/opt/OPSWGw/gateway-name/opswgw.log*
```

Global File System Server Logs

These logs are in the following files:

```
/var/lc/OPSWHub/OPSWHub.log*
/var/opt/OPSWogfs/*err
```

Global Shell Audit Logs

When a user accesses or modifies a managed server with the Global Shell feature, Opware SAS records the event in an audit log. The Global Shell audit logs contain information about the following events:

- Logins and logouts with Global Shell and Remote Terminal sessions
- File system operations (such as create and remove) on managed servers
- Commands and scripts that run on managed servers through the Remote Opware Shell (`rssh`)



The Global Shell audit logs are on the server where the Opware Global File System (OGFS) is installed.

To view a log file, open a terminal window, log into the server running the OGFS, and use a command-line utility such as `more`, `grep`, or `vi`. The Global Shell audit logs are made up of three sets of logs files:

- Shell event logs
- Shell stream logs
- Shell script logs

Shell Event Logs

The shell event logs contain information about operations that users have performed on managed servers with the Global Shell. These logs are in the following directory (where `ogfs-host` is the name of the server running the OGFS):

```
/var/opt/OPSWmnt/audit/event/ogfs-host
```

The log file name has the following syntax (where `n` is the log rotation number):

```
audit.log.n
```

For each event, Opsware SAS writes a single line to an event log file. Each line in the log file contains the following information about the event:

- Unique ID of the event
- Unique ID of the parent event
- Date of the operation
- ID of the Opsware user who performed the operation
- Name of the Opsware user who performed the operation
- Name of the component that generated the audit event
- Version of the Opsware SAS component that generated the audit event
- Name of the Opsware SAS feature which generated the audit event
- Name of the operation (action)
- Verbosity level
- Exit status of the event
- ID of the managed server
- Name of the managed server
- Details of the event

The following example shows a single line in an audit event log file:

```
jd@e@m185:051202182224813:13  jd@e@m185:051202182224790:12
2006/01/28-12:40:19.622 User.Id=2610003 User.Name=jd@e
Hub:1.1 GlobalShell      AgentRunTrustedScript    1      OK
Device.Id=10003 Device.Name=m192.dev.opsware.com
ConnectMethod=PUSH      RemotePath=      RemoteUser=root
ScriptName=__global__.sc_snapshot.sh
ScriptVersion=30b.2.1572  ChangeTime=1128971572
RemoteErrorName=
```

In this example, the first field is the ID of the event:

```
jd@e@m185:051202182224813:13
```

This ID field has the following syntax:

```
opsware-user@ogfs-host:YYMMDDHHmmssSSS:n
```

The ID field matches the name of a shell stream log file.

Shell Stream Logs

The shell stream logs contain the `stdout` of scripts that are run from the Global Shell. These logs are in the following directory (where `ogfs-host` is the name of the server running the OGFS):

```
/var/opt/OPSWmnt/audit/streams/ogfs-host
```

The log file name has the following syntax:

```
opsware-user@ogfs-host:YYMMDDHHmmssSSS:n
```

The log file name matches the ID field in the shell event log. A header line in the log file contains the file name, character set, version, and Opware user name. If the `stdout` of the script contains control characters, the shell stream log will contain the same control characters.

Shell Script Logs

The shell script logs contain the contents of scripts that are run from the Global Shell. These logs are in the following directory (where `ogfs-host` is the name of the server running the OGFS):

```
/var/opt/OPSWmnt/audit/scripts/ogfs-host
```

The log file name is a hash string based on the script contents, for example:

```
23f1d546cc657137fa012f78d0adfdd56095c3b5
```

A header line in the log file contains the file name, character set, version, and Opware user name.

Digital Signatures in the Global Shell Audit Logs

The shell stream and script log files contain digital signatures and fingerprints, which are generated with the RSA-SHA1 algorithm. To verify the signature and fingerprint of a log file, open a terminal window, log into the OGFS server, and enter the following command:

```
/opt/OPSWagentproxy/bin/auditverify stream_file_name \  
rsa_key_path
```

Here's an example in bash:

```
STREAMDIR=/var/opt/OPSWmnt/audit/streams/somehost.opsware.com  
STREAMFILE=jdoe@somehost:051210003000111:61  
RSAKEYPATH=/var/lc/crypto/cogbot/cogbot.srv
```

```
/opt/OPSWagentproxy/bin/auditverify $STREAMDIR/$STREAMFILE \  
$RSAKEYPATH
```

If the log file has not been tampered with, `auditverify` displays the following message:

```
[AuditVerify]: Verification Result: Valid Signature
```

By default, the logs are signed with the private key in the following file:

```
/var/1c/crypto/cogbot/cogbot.srv
```

To change the key file used for signing, modify the `audit.signature.key_path` parameter in the System Configuration page of the Opware Command Center. For instructions on accessing the System Configuration page, see “Configuring the Global Shell Audit Logs” on page 136.

Storage Management for the Global Shell Audit Logs

By periodically removing the shell stream and script log files, Opware SAS prevents these files from filling up the available disk space. The System Configuration page of the Opware Command Center contains parameters that determine when the log files are removed. These parameters enable you to specify the removal of the log files based on the age (`archive_days`) of the files or the amount of disk space (`archive_size`) used by the files.

The following parameters specify the age of the files to remove:

```
audit.stream.archive_days
audit.script.archive_days
```

The following parameters specify the amount of disk space that the files can occupy before they are removed:

```
audit.stream.archive_size
audit.script.archive_size
```

For details on these parameters, see Table 4-3. For instructions on accessing the System Configuration page of the Opware Command Center, see “Configuring the Global Shell Audit Logs” on page 136.

Table 4-3: Parameters for Global Shell Audit Log Configuration

PARAMETER	DESCRIPTION	DEFAULT VALUE
<code>audit.root.dir</code>	The root directory for audit streams and scripts.	<code>/var/opt/OPSWmnt/audit/</code>

Table 4-3: Parameters for Global Shell Audit Log Configuration (continued)

PARAMETER	DESCRIPTION	DEFAULT VALUE
audit.script.archive_days	Audit script files older than this value (in days) are deleted. 0 means files are never deleted.	100
audit.script.archive_size	Maximum amount of disk space (in MB) used by all audit script files. Older files are removed first. 0 means no maximum.	100
audit.signature.algorithm	Signature algorithm to use when signing audit streams.	RSA-SHA1
audit.signature.key_path	Location of the private key used when signing audit streams.	/var/lc/crypto/cogbot/ cogbot.srv
audit.stream.archive_days	Audit stream files older than this value (in days) are deleted. 0 means files are never deleted.	10
audit.stream.archive_size	Maximum amount of disk space (in MB) used by all audit stream files. Older files are removed first. 0 means no maximum.	1000
audit.stream.file_keep	Maximum number of rotated audit stream files.	50
audit.stream.file_size	Maximum file size for audit streams. Specified in MB. The largest allowed value is 50MB.	10

Configuring the Global Shell Audit Logs

You can change parameters such as the maximum log file size. For a list of the parameters, see Table 4-3 on page 134. To configure the parameters, perform the following steps:

- 1** In the Opware Command Center, under Administration click the System Configuration link.
- 2** On the “System Configuration: Select Product” page, click the hub link.
- 3** On the “System Configuration: Set Configuration Parameters” page, you can change parameters such as audit.root.dir.
- 4** Click **Save**.

Overview of Restarting Opware Components

This section provides information on restarting Opware components and contains the following topics:

- Restarting Opware Components
- Restarting the Boot Server
- Restarting the Build Manager
- Restarting the Command Engine
- Restarting the Data Access Engine
- Restarting the Media Server
- Restarting the Model Repository
- Restarting the Model Repository Multimaster Component
- Restarting an Opware Agent
- Restarting the Opware Command Center
- Restarting the Software Repository
- Restarting the Software Repository, Multimaster Component
- Restarting the Web Services Data Access Engine
- Restarting the Opware Gateway

- Restarting the Global File System Server

Restarting Opware Components

While maintaining Opware SAS, you might encounter operational problems or an Opware component might fail. A server might become unreachable because the Opware Agent on the server has stopped responding.

Opware SAS components are installed on both Linux and Solaris servers. The processes run in the same directories on both platforms.

As part of maintaining Opware SAS, you might need to stop and restart specific components or the Opware Agent on an Opware-managed server.

Opware components are configured to restart upon process failure, meaning that if a component's process terminates, nothing needs to be done to restart that process. However, there might be times when a manual restart is necessary. There are some restart ordering dependencies to take into consideration when you manually restart Opware components.

You must restart the following components in the following order:

- 1** Make sure that the machine on which the Model Repository resides is running, and that Oracle is also running.
- 2** If you restart the Model Repository, you must also restart the Data Access Engine and, in a multiple facility environment, restart the Model Repository Multimaster Component.
- 3** If you need to restart the Web Services Data Access Engine, the Data Access Engine, and the Model Repository must be running first.
- 4** If you need to restart Build Manager, these components must be started first: the Data Access Engine, the Command Engine, and the Web Services Data Access Engine.
- 5** In a multiple facility environment, if you need to stop and restart the TIBCO Rendezvous process (`rvrđ`): follow this procedure:

Stop the Opware components that are running on the same host. If you leave the components running, they spawn a new `rvđ` (not `rvrđ`) process.

Stop the `rvrđ` process: `/etc/init.d/rvrđscript stop`

Start the `rvrđ` process.

Start the Opware components that are running on the same host.

- 6** In a multiple facility environment, if you need to restart the Web Services Data Access Engine, in addition to the Data Access Engine, and the Model Repository, you must also be sure that the Model Repository Multimaster Component is running first.

Restarting the Boot Server

You should not need to restart the Boot Server. However, you might need to restart the DHCP server that runs on the Boot Server.

To stop the DHCP server, enter the following command as root at the prompt on the server running the Boot Server:

```
/etc/init.d/dhcpd stop
```

To start the DHCP server, enter the following command as root at the prompt on the server running the Boot Server:

```
/etc/init.d/dhcpd start
```

Restarting the Build Manager

The Build Manager was installed when the Build Scripts ran during installation. To stop the Build Manager, enter the following command as root at the prompt on the server running the Build Manager:

```
/etc/init.d/buildmgr stop
```

To start the Build Manager, enter the following command as root at the prompt on the server running the Build Manager:

```
/etc/init.d/buildmgr start
```

Restarting the Command Engine



Use caution when stopping and restarting the Command Engine. If processes are active when the Command Engine stops, data might be lost.

Before you stop the Command Engine, check the Way Administration page:

```
https://[hostname:1018]/way/bidniss/activeScripts.py
```

If none of the processes has a status of Active, it is safe to stop the Command Engine. There are also links to Command Engine processes at the following URL:

```
https://[hostname:1018]/way/bidniss/
```



To access the Way Administration page, you must have an Opware certificate of authority. Contact Opware Support.

Click the Write Machine link. If the resulting table shows messages in the Write Queue column, there are active processes, and stopping the Command Engine results in a loss of data.

To stop the Command Engine, enter the following command as root at the prompt on the server running the Command Engine:

```
/etc/init.d/waybot stop
```

To start the Command Engine, enter the following command as root at the prompt on the server running the Command Engine:

```
/etc/init.d/waybot start
```

Restarting the Data Access Engine

To stop the Data Access Engine, enter the following command as root at the prompt on the server running the Data Access Engine:

```
/etc/init.d/spin stop
```

To start the Data Access Engine, enter the following command as root at the prompt on the server running the Data Access Engine:

```
/etc/init.d/spin start
```

Restarting the Media Server

For Windows, check to see if the Media Server process is running by entering the following command as root at the prompt on the server running the Windows Media Server:

```
ps -e|grep -w smbd
```

To stop the Windows Media Server, enter the following command as root at the prompt on the server running the Windows Media Server:

```
/etc/init.d/samba.server stop
```

To start the Windows Media Server, enter the following command as root at the prompt on the server running the Windows Media Server:

```
/etc.init.d/samba.server start
```

To restart the Windows Media Server, enter the following command as root at the prompt on the server running the Windows Media Server:

```
/etc/init.d/samba.server restart
```

Restarting the Model Repository

To make sure that the Oracle database is running on Solaris and Linux, enter the following command:

```
ps -fu oracle | grep pmon
```

To make sure that the Oracle listener is running on Solaris, enter the following command:

```
ps -fu oracle | grep -i listen
```

To make sure that the Oracle listener is running on Linux, enter the following command:

```
ps axwww |grep -i listen
```

In both cases, the correct output is one or more lines, and no output at all is an error.

To stop and restart the Model Repository, contact your company database administrator.

Restarting the Model Repository Multimaster Component

The Model Repository, Multimaster Component is automatically installed on the same server as the Model Repository. To stop the Model Repository, Multimaster Component, enter the following command as root at the prompt on the server running the Model Repository:

```
/etc/init.d/vaultdaemon stop
```

To start the Model Repository, Multimaster Component, enter the following command as root at the prompt on the server running the Model Repository:

```
/etc/init.d/vaultdaemon start
```



Before you restart the Model Repository, Multimaster Component, perform these verification steps: Access the Model Repository, Multimaster Component logs to determine if the Model Repository, Multimaster Component shut down. When shutdown, the Model Repository, Multimaster Component logs contain the statement, "Vault has been shut down." Check that the server running the Model Repository, Multimaster Component has stopped the Model Repository, Multimaster Component process.

Restarting an Opware Agent

To stop the Opware Agent on the Solaris platform, execute the command:

```
/etc/init.d/cogbot stop
```

To restart the Opware Agent on the Solaris platform, execute the command:

```
/etc/init.d/cogbot start
```

For Linux 6.2, to stop the Opware Agent, execute the command:

```
/etc/rc.d/init.d/cogbot stop
```

For Linux 6.2, to restart the Opware Agent, execute the command:

```
/etc/rc.d/init.d/cogbot start
```

For Linux 7.2 and higher, to stop the Opware Agent, execute the command:

```
/etc/init.d/cogbot stop
```

For Linux 7.2 and higher, to restart the Opware Agent, execute the command:

```
/etc/init.d/cogbot start
```

For AIX, to stop the Opware Agent, execute the command:

```
/etc/rc.d/init.d/cogbot stop
```

For AIX, to restart the Opware Agent, execute the command:

```
/etc/rc.d/init.d/cogbot start
```

For HP-UX, to stop the Opware Agent, execute the command:

```
/sbin/init.d/cogbot stop
```

For HP-UX, to restart the Opware Agent, execute the command:

```
/sbin/init.d/cogbot start
```

On Windows, enter the following commands:

```
net stop shadowbot  
  
net start shadowbot
```

Restarting the Opware Command Center

To stop the Opware Command Center, enter the following command as root at the prompt on the server running the Opware Command Center:

```
/etc/init.d/owm.server stop
```

To start the Opware Command Center, enter the following command as root at the prompt on the server running the Opware Command Center:

```
/etc/init.d/owm.server start
```

Restarting the Software Repository

To stop the Software Repository, enter the following commands as root at the prompt on the server running the Software Repository:

```
/etc/init.d/mm_wordbot stop  
  
/etc/init.d/mm_wordbot-clear stop
```

To start the Software Repository, enter the following commands as root at the prompt on the server running the Software Repository:

```
/etc/init.d/mm_wordbot start  
  
/etc/init.d/mm_wordbot-clear start
```

Restarting the Software Repository, Multimaster Component

To stop the Software Repository, Multimaster Component, enter the following command as root at the prompt on the server running the Software Repository:

```
/etc/init.d/mmworddaemon stop
```

To start the Software Repository, Multimaster Component, enter the following command as root at the prompt on the server running the Software Repository:

```
/etc/init.d/mmworddaemon start
```

Restarting the Web Services Data Access Engine

The Web Services Data Access Engine was installed automatically with the Opware Command Center during installation, and it runs on the same server. To stop the Web Services Data Access Engine, enter the following command as root at the prompt on the server running the Opware Command Center:

```
/etc/init.d/twist stop
```

To start the Web Services Data Access Engine, enter the following command as root at the prompt on the server running the Opware Command Center:

```
/etc/init.d/twist start
```



The Web Services Data Access Engine start is asynchronous. It could take up to ten minutes to restart, so the Opware Command Center is unavailable during that time.

Restarting the Opware Gateway

To stop the Opware Gateway, enter the following command as root at the prompt on the server running the Opware Gateway:

```
/etc/init.d/opswgw-<NAME> stop
```

To start the Opware Gateway, enter the following command as root at the prompt on the server running the Opware Gateway:

```
/etc/init.d/opswgw-<NAME> start
```

The <NAME> is the name of the Opware Gateway.

Restarting the Global File System Server

To stop the Global File System Server, enter the following command as root at the prompt on the server running the Global File System Server:

```
/etc/init.d/OSPWhub stop
```

To start the Global File System Server, enter the following commands as root at the prompt on the server running the Global File System Server:

```
/etc/init.d/OSPWhub start
```

Opsware Software

The Opsware Software function is populated during Opsware SAS installation.

Each component of Opsware SAS is shown by its internal name, and the display is graphically similar in appearance to the way that nodes appear in the Software Tree. You cannot add or delete components or nodes in this area of Opsware SAS.

Table 4-1 shows the internal and external names of Opsware SAS components.

Table 4-1: Opsware Internal and External Component Names

INTERNAL NAME	EXTERNAL NAME
Agent	Opsware Agent
buildmgr	OS Build Manager
hub	Global File System Server
occ	Opsware Command Center
spin	Data Access Engine
truth	Model Repository
twist	Web Services Data Access Engine
vault	Model Repository Multimaster Component
way	Command Engine
word	Software Repository

Some of the functionality available in the Server Management area of the system is also available to be applied to the servers that appear on the Members tab. Take care in applying changes to the core servers. In particular, do not assign or unassign servers to these nodes or install or uninstall software or change networking unless directed to do so during the installation process by the *Opsware® SAS Deployment and Installation Guide*.

To view the servers on which each component is installed, click the component's hyperlinked name, then select the Members tab. The number of servers associated with that component appears on the tab itself, and detailed information about those servers shows when you select the tab.

Mass Deletion of Backup Files

Opware SAS includes a script that you can run as a cron job for performing mass deletions of backup files. Backup files are created by configuration tracking. They can accumulate quickly and take up disk space. Consequently, performance when viewing backup history in the Opware Command Center can be sluggish, and the information that displays might be cluttered with out-of-date configuration tracking data.

When the backup deletion script is run, it deletes all backed up files with the exception that it always keeps one copy of the latest version of every file ever backed up. If you want to delete those files, use the process for deleting backups individually or a few at a time that is covered in the *Opware® SAS User's Guide*.

The script is called `backup_delete.pyc`. It is located on the server where the Data Access Engine resides, in the following directory:

```
/cust/usr/blackshadow/spin/util
```

The script is run using a configuration file that contains the script arguments such as host name, port number, whether you want full or incremental backups, the backup retention period, the name of the log file to use, email addresses for notifications, and the email server to use. See Table 4-2, Configuration File Options, for the arguments, their values, and their descriptions.

Command Syntax

```
backup_delete.pyc [options]
```

```
Usage: backup_delete.py [-c <conf_filename>]
```

Deleting Backup Files with the Mass Deletion Script

Perform the following steps to use the mass deletion script to delete backup files:

- 1** Log in as root to the server where the Data Access Engine is installed.
- 2** Make sure that `/lc/blackshadow` is in your `PYTHONPATH` environment variable.
- 3** Create a file that contains the arguments and values that you want Opware SAS to use with the mass deletion script. See Table 4-2 on page 146, Configuration File Options, for the available arguments.

For example, the following file specifies that a host called `spin.yourcore.example.com`, on port 1004 will have incremental backups that are three months old deleted. In addition, a log file called `run.log`, located in `/tmp`

will be used to capture events, and email will be sent to `user@example.com` from `user1@example.com` reporting that the mass deletion was performed successfully.

```
host: spin.yourcore.example.com
port: 1004
inc: 1
time: 3m
logfile: /tmp/run.log
emailto: user@example.com
emailserver: smtp.example.com
emailfrom: user1@example.com
emailsucces: 1
```

Table 4-2: Configuration File Options

ARGUMENTS	VALUES	DESCRIPTION
host	host: [hostname], for example host: spin.yourcore.example.com	Host name of the Data Access Engine
port	port: [port number], for example port: 1004	Port of the Data Access Engine (defaults to 1004)
full	Set value to 1 to enable, for example full:1	Delete full backups. You must specify Either full or inc.
inc	Set value to 1 to enable, for example inc:1	Delete incremental backups. You must specify either full or inc.
time	time: [digits] [dmy], for example, 6d equals six days. 3m equals three months. 1y equals one year.	Retention period beyond which backups should be deleted.

Table 4-2: Configuration File Options

ARGUMENTS	VALUES	DESCRIPTION
hostsfile	hostsfile: [filename] The hostsfile should contain the name of each host on a line by itself, for example <hostname> <hostname>	The script deletes backups on every managed server in your system, unless you provide a hostsfile that contains a specific list of servers on which to perform the mass backup deletion.
logfile	logfile: [filename], for example logfile: /tmp/ run.log	File to use for log events.
emailto	emailto: [email address], for example emailto: user@example.com	Optional email notification recipient.
emailserver	emailserver: [server name], for example emailserver: smtp.example.com	The SMTP server to send email through. Optional if emailto not specified, otherwise required.
emailfrom	emailfrom: [email address], for example emailfrom: user1@example.com	Email address to appear in the From: line. Optional if emailto not specified, otherwise required.
emailsucces	Set value to 1 to enable, for example emailsucces: 1	Send email even if no errors occurred deleting backups and more than one backup was deleted.

- 4** Optionally, if you want to run the script as a cron job, create a crontab entry.

For example, to run the job at 3:00 AM daily, create the following entry:

```
0 3 * * * env PYTHONPATH=/lc/blackshadow /lc/bin/python/  
cust/usr/blackshadow/spin/util/backup_delete.pyc -c <path>/  
<your_backup_filename.conf>
```



The crontab entry must be all on one line.

- 5** If you do not plan to run the script as a cron job, enter the following command at the prompt:

```
# python /cust/usr/blackshadow/spin/util/backup_delete.pyc \  
c / [conf_filename]
```

Designations for Multiple Data Access Engines

This section discusses the following topics:

- Overview of Designations for Multiple Data Access Engines
- Reassigning the Data Access Engine to a Secondary Role
- Designating the Multimaster Central Data Access Engine

Overview of Designations for Multiple Data Access Engines

In a core with multiple instances of the Data Access Engine, each instance may be designated in one of the following ways:

- **Primary Data Access Engine:** Each facility has only one primary Data Access Engine. This Data Access Engine periodically checks the managed servers to determine if Opsware SAS can communicate with them. If a facility has more than one primary Data Access Engine, the competing reachability checks can interfere with each other.
- **Secondary Data Access Engine:** When a facility has multiple Data Access Engines installed (for scalability), the additional ones are designated secondary. The first Data Access Engine installed is designated the Primary or Multimaster Central Data Access Engine. A secondary Data Access Engine does not check managed servers to determine if they are reachable. It only communicates with the Model Repository write or read data.
- **Multimaster Central Data Access Engine:** An Opsware multimaster mesh of cores has only one multimaster central Data Access Engine. Although any of the cores may

have multiple Data Access Engines, only one engine in the mutilmaster mesh can be the central engine.

Reassigning the Data Access Engine to a Secondary Role

If you installed an additional Data Access Engine, you must perform the following steps to reassign the new Data Access Engine to a secondary role:

- 1** Log into the Opware Command Center as a user that belongs to Opware SAS Administrators group.

The Opware Command Center should be installed and listening. The Opware Command Center home page appears.
- 2** Click Administration ► Opware Software from the navigation panel. The Opware Software page appears.
- 3** Click the spin link. The Opware Software | spin page appears.
- 4** Select the Members tab. The list of servers that are running the Data Access Engine in the core appears.
- 5** Select the check box for the additional Data Access Engine server.
- 6** From the **Tasks** menu, select **Re-Assign Node**.
- 7** Select the option for the Service Levels | Opware | spin node.
- 8** Click **Select**.
- 9** Navigate the node hierarchy by clicking the following nodes:
 - Opware
 - spin
 - Secondary
- 10** Click **Re-Assign**.
- 11** In a terminal window, log in as root to the server running the additional Data Access Engine and enter the following command to restart the Data Access Engine:

`/etc/init.d/spin restart`

Designating the Multimaster Central Data Access Engine

The Opsware Installer automatically assigns the multimaster central Data Access Engine.



Opsware, Inc. recommends that you do not change the multimaster central Data Access Engine after the installation. Doing so might cause problems when upgrading the Opsware core to a new version. Before following the steps in this section, contact your Opsware, Inc. support representative

Perform the following steps to designate the multimaster central data access engine:

- 1** Log into the Opsware Command Center as a user that belongs to the Opsware System Administrators group.
- 2** From the navigation panel, click Opsware Software under Administration. The Opsware Software page appears.
- 3** Click the spin link.
- 4** Select the Servers tab.
- 5** Select the check box for the Data Access Engine server for the new core.
- 6** From the **Server** menu, select **Re-Assign Node**.
- 7** Select the option for the Service Levels | Opsware | spin | node.
- 8** Click **Select**.
- 9** Navigate the node hierarchy by clicking each node: Opsware | Spin | Multimaster Central.
- 10** Click **Re-Assign**.
- 11** Restart the Multimaster Central Data Access Engine.

```
/etc/init.d/spin restart
```

Web Services Data Access Engine Configuration File

The Web Services Data Access Engine configuration file contains properties that affect the server side of the Opware SAS Web Services API. The name of the configuration file is `twist/etc/twist.conf`. The location of the file is relative to the installation directory. For example, if the Web Services Data Access Engine is installed on `/cust`, the configuration file is located in the following directory:

```
/cust/twist/etc/twist.conf
```



During an upgrade, `twistOverrides.conf` file is created in addition to the `twist.conf` file. Every time you upgrade to a new version, the `twist.config` file may change, but the `twistOverrides.conf` file is preserved between upgrades. When you upgrade to a new version of SAS, to make changes to the properties defined in the configuration file, you must edit the `/cust/twist/etc/twistOverrides.conf` file.

To change a property defined in the configuration file:

- 1** Edit the file with a text editor.
- 2** Save the changed file.
- 3** Restart the Web Services Data Access Engine on the server.



You must be an Opware administrator in order to modify the `twist.conf` file. Once the file is changed, the Web Services Data Access Engine must be restarted to institute the changes.

The following table lists the properties of the configuration file. Several of these properties are related to the cache (sliding window) of server events. Opware SAS maintains a sliding window (with a default size of two hours) of events describing changes to `Server` objects. This window makes it possible for software developers to update a client-side

cache of `Server` objects without having to retrieve all of the objects. For more information, see the `getEventList` operation of the Server Web Service in the *Opsware® SAS Web Services API Guide*.

Table 4-3: Web Services Data Access Engine Configuration File

PROPERTY	DEFAULT	DESCRIPTION
<code>twist.webservices.debug.level</code>	1	An integer value that sets the debug level for the Opsware Web Services API on the server side. Allowed values: 0 - basic info 1 - more detailed information 2 - stack trace 3 - for printing the server event cache entries whenever there is an item added to the cache.
<code>twist.webservices.locale.country</code>	US	The country Internationalization parameter for the Localizer utility. Currently only the <code>US</code> code is supported.
<code>twist.webservices.locale.language</code>	en	Sets the language Internationalization parameter for the Localizer utility. Currently only the <code>en</code> code is supported.
<code>twist.webservices.caching.windowsize</code>	120	In minutes, the size of the sliding window maintaining the server event cache.
<code>twist.webservices.caching.windowslide</code>	15	In minutes, the sliding scope for the window maintaining the server event cache.

Table 4-3: Web Services Data Access Engine Configuration File

PROPERTY	DEFAULT	DESCRIPTION
<code>twist.webservices.caching.safetybuffer</code>	5	In minutes, the safety buffer for the sliding window maintaining the server event cache.
<code>twist.webservices.caching.minwindowsize</code>	30	In minutes, the minimum size of the sliding window that maintains the server event cache.
<code>twist.webservices.caching.maxwindowsize</code>	240	In minutes, the maximum size of the sliding window that maintains the server event cache.

Adding Locales to the Opware Command Center Component

For the OCC Client to display multi-byte characters correctly, the Opware Command Center component must have the correct locale preferences.

To add a locale preference, perform the following steps.

- 1** On the core server that runs the Opware Command Center component, open the following file in a text editor:
`/cust/occ-version/etc/psrvr.properties`
- 2** In this properties file, add the locale to the following line:
`pref.user.localesAllowed=en;ja;`
For example, the following line includes the Korean locale:
`pref.user.localesAllowed=en;ja;ko;`
- 3** Save the properties file and exit the editor.
- 4** Restart the component. See “Restarting the Opware Command Center” on page 142 in this chapter for more information.
- 5** If your core has multiple Opware Command Center components, perform the preceding steps on each core server that runs an Opware Command Center component.

Automatically Importing Windows Patches

Microsoft posts patches on its web site on the second Tuesday of each month, unless a special circumstance requires an immediate release. Before Opware SAS can install a patch on a managed server, the patch must be downloaded from the Microsoft web site and imported (uploaded) into the Software Repository. You can download and import patches with either the OCC Client or with the script described in this section. For information on importing patches with the OCC Client, see the *Opware® SAS User's Guide*.

The `populate-opware-update-library` shell script downloads and imports both patches and the Microsoft Patch Database. (To be imported, a patch must be in the Microsoft Patch Database that has been imported into the Software Repository.) Optionally, the script sets the initial status (Available or Limited) of newly imported patches. The script can also filter the patches imported according to operating system (such as Windows NT).

You can schedule the `populate-opware-update-library` script to run periodically as a `cron` job on the Software Repository server. To end-users of the OCC Client, the patches imported with the script appear to have been automatically imported. Do not run concurrent instances of the script.

The `populate-opware-update-library` script is in the following directory:

```
/cust/usr/blackshadow/mm_wordbot/util/
```

Table 4-4 lists the script's options. The `--set_available` and `--set_limited` options cannot be specified at the same time.

Table 4-4: Options of `populate-opware-update-library`

OPTION	DESCRIPTION
<code>--spin hostname-or-IP</code>	Hostname or IP address of Data Access Engine (spin) host. Default value: spin
<code>--theword hostname-or-IP</code>	Hostname or IP address of Software Repository (theword) host. Default value: theword

Table 4-4: Options of `populate-opware-update-library` (continued)

OPTION	DESCRIPTION
<code>--cert_path file-path</code>	File specification of cert file to be used for Spin connection. Default value: <code>/var/lc/crypto/wordbot/wordbot.srv</code>
<code>--ca_path file-path</code>	File specification of CA file to be used for Spin connection. Default value: <code>/var/lc/crypto/wordbot/opware-ca.crt</code>
<code>--verbose</code>	Display copious output.
<code>--no_nt4</code>	Do not process NT4 patches.
<code>--no_w2k</code>	Do not process W2K patches.
<code>--no_w2k3</code>	Do not process W2K3 patches.
<code>--use_proxy_url url</code>	When downloading binaries, connect via this proxy URL.
<code>--proxy_userid userid</code>	Basic-auth userid to provide to proxy server.
<code>--proxy_passwd passwd</code>	Basic-auth passwd to provide to proxy server.
<code>--set_available</code>	Set availability status to Available when uploading patches.
<code>--set_limited</code>	Set availability status to Limited when uploading patches.
<code>--no_hotfixes</code>	Do not upload hotfixes.
<code>--no_servicepacks</code>	Do not upload servicepacks.
<code>--no_updaterollups</code>	Do not upload updaterollups.
<code>--no_mssecure_upload</code>	Do not upload the MBSA 1.2 patch database.
<code>--no_wsusscan_upload</code>	Do not upload the MBSA 2.0 patch database.

Table 4-4: Options of populate-opware-update-library (continued)

OPTION	DESCRIPTION
<code>--wsusscan_url_override url</code>	Download the MBSA 2.0 patch database from this URL.
<code>--mssecure_url_override url</code>	Download the MBSA 1.2 patch database from this URL.
<code>--update_all</code>	Refresh the patches already uploaded into Opware SAS.
<code>--download_only path</code>	Download files from the vendor's web site to the specified path, but do not upload them into Opware SAS.
<code>--help</code>	Display the syntax of this script.

Index

A

- accessing, realm information 99
- administrators. *See* Opsware administrators.
- agent-server architecture 23
- Application Configuration Management, overview . 57
- auditverify tool 133

B

- backup, deleting files 145
- Boot Server
 - defined 23, 26
 - logs 127
 - restarting 138
- Build Agent, defined 23, 29
- Build Manager
 - defined 23, 26
 - logs 128
 - restarting 138

C

- CDR. *See* Code Deployment & Rollback.
- Command Engine
 - defined 22, 23
 - logs 128
 - restarting 138
 - scripts 26
 - system diagnostic tests 125
- conflicts
 - alert emails 88
 - causes 70
 - error messages 89
 - overview 69
 - prevention 73
 - resolving 79, 84
- contacting, Opsware support 120
- content management, tools 46
- conventions used in the guide 12
- creating, Manual updates 113

D

- Data Access Engine
 - defined 23, 26
 - logs 128
 - multiple 148
 - reassigning 149
 - restarting 139
 - See also* Multimaster Central Data Access Engine.
 - system diagnostic tests 123
- deleting, backup files 145
- diagnosing, problems 121
- digital 133
- dormant, Opsware Agents 28

E

- enabling, realm information 97

F

- facilities
 - defined 95
 - definition of 22
 - multiple 65
 - primary 69
 - viewing, information 97

I

- Inbound, Model Repository Multimaster Component . 27
- installations
 - multiple Data Access Engine 148
 - types 22
- installing
 - patch 38
 - scripts 43
- integrating, Opsware SAS with AIX and HP-UX ... 51

J

- JBoss 129

L

locale	153
log	
digital signatures	133
logs	
about	127
Boot Server	127
Build Manager	128
Command Engine	128
configuring	129, 136
Data Access Engine	128
Global Shell Audit	131
JBoss	129
managed servers	
Global Shell logs	131
Media Server	128
Model Repository	128
Model Repository Multimaster Component	128
Opware Agents	129
Opware Command Center	129
Software Repository	129
Software Repository Multimaster Component	129
Software Repository Replicator	129
Web Services Data Access Engine	130

M

Manual updates	
creating	113
defined	107
overview	111
Software Repository Cache, applying to	115
uploading, Microsoft utilities	116
Media Server	
defined	23, 26
logs	128
restarting	139
Model Repository	
defined	21, 23, 27
logs	128
restarting	140
Model Repository Multimaster Component	
defined	23, 27
Inbound	27
logs	128
Outbound	27
restarting	140
system diagnostic tests	125
model-based control	21
multimaster	
alert emails during conflicts	88

central	69
conflicts	69
designating the Central Data Access Engine	150
error messages in multimaster conflicts	89
installation	22
mesh	69, 74
mode	69
network administration	88
preventing conflicts	73
tools	74
multimaster central	69
Multimaster Central Data Access Engine	150
multimaster, tools	74, 78

N

network administration	88
------------------------------	----

O

On-demand updates	
defined	107
overview	111
Opware administrators, installing scripts	43
Opware Agent	
defined	22, 23
dormant	28
Installer	28
logs	129
overview	27
restarting	141
Opware Agent Installer	28
Opware Command Center	
defined	21, 23, 28
logs	129
restarting	142
Opware components	
internal and external names	144
overview	25
restarting	137
running, system diagnosis	126
Opware Discovery and Agent Deployment, overview	
55	
Opware Gateway	
defined	24, 30
Opware Global File System	
defined	31
Opware guides	
contents	11
conventions used	12
documentation set	14

icons in guide, explained	13
Opware SAS	
agent-server architecture	23
components	25
components overview	23
core technology	20
documentation set	14
environment	20
integrating with AIX and HP-UX	51
model-based control	21
multimaster mode	69
multiple facilities	65
overview	17
related documentation	14
restarting Opware components	137
security	32
software provisioning	43
system diagnosis	121
tools	46
troubleshooting	120
types of users	19
Opware Satellite	
accessing, realm information	99
definition	22
linked to cores	22
manual update	107
on-demand updates	107
overview	93
permissions, required	96
Software Repository Cache, overview	105
OS Build Agent. See Build Agent.	
Outbound, Model Repository Multimaster Component	27

P

patch management	
installing, patches	38
Microsoft Patch Database	154
Microsoft patch releases	154
uninstalling, patches	40
updating, Microsoft patch	42
uploading automatically	154
uploading, patches	37
populate-opware-update-library	154
preventing, conflicts	73
preview reconcile	44
Primary Data Access Engine	148
primary facility	69
Python	26

R

realms	
defined	95
enabling realm information	97
viewing realm information	99
reassigning, Data Access Engine	149
reconcile	45
reconciling	29
resolving	
conflicts by object	79
conflicts by transaction	84
restarting	
Boot Server	138
Build Manager	138
Command Engine	138
Data Access Engine	139
Media Server	139
Model Repository	140
Model Repository Multimaster Component	140
Opware Agents	141
Opware Command Center	142
Opware components	137
Software Repository	142
Software Repository Multimaster Component	142
Web Services Data Access Engine	143
running, system diagnosis	126

S

Satellite. See Opware Satellite.	
scripts	
Command Engine	26
deleting backup files	145
Secondary Data Access Engine	148
software provisioning	
overview	43
preview reconcile	44
reconcile	45
Software Repository	
defined	23, 29
logs	129
mapping	32
restarting	142
system diagnostic tests	124
Software Repository Cache	
applying, Manual updates	115
defined	24, 30
managing	105
overview	112
packages, availability of	106
staging files	116

Opware Inc. Confidential Information: Not for Redistribution. Copyright © 2000-2006 Opware Inc. All Rights Reserved.

159

Software Repository Multimaster Component		defined	24, 30
conflicts	69	logs	130
defined	23	restarting	143
logs	129	system diagnostic tests	124
restarting	142		
Software Repository Replicator			
defined	23, 29		
logs	129		
Software Repository, Multimaster Component			
defined	30		
standalone installation	22		
system diagnosis			
Command Engine tests	125		
contacting, support	120		
Data Access Engine tests	123		
diagnosing, problems	121		
Model Repository Multimaster Component tests	125		
running, system diagnosis	126		
Software Repository tests	124		
testing	122		
troubleshooting, problems	120		
Web Services Data Access tests	124		
system diagnosis, tools	121, 122, 126		

T

tools	
content management tools	46
multimaster tools	74, 78
system diagnosis	121, 122, 126
troubleshooting	120

U

uninstalling, patch	40
updating, Microsoft patch	42
uploading, patch	37
users, of Opware	19

V

viewing	
facilities information	97
realm information	99
visual packager, overview	59

W

Web Services Data Access Engine	
configuration file	151