



Opsware[®] SAS 6.5 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 6.5

Copyright © 2000-2007 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Opware Inc. Unpublished Confidential Information.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, SAS Web Client, 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/sas65tpos.pdf>.

Table of Contents

Preface	15
<hr/>	
Overview of this Guide	15
Contents of this Guide	15
Conventions in this Guide	16
Icons in this Guide	17
Guides in the Documentation Set and Associated Users	17
Opware, Inc. Contact Information	18
Chapter 1: Opware SAS Overview	21
<hr/>	
Opware SAS Technology	21
Types of Opware Users	23
Opware SAS Environment	24
Model-Based Control	25
Types of Opware SAS Installations	26
Opware SAS Components	27

Boot Server30
Build Manager30
Command Engine30
Data Access Engine30
Media Server30
Model Repository31
Model Repository Multimaster Component31
Opware Agents31
Dormant Opware Agents32
Opware SAS Client32
Opware SAS Web Client32
Opware Command Center33
OS Build Agent33
Software Repository33
Software Repository Replicator33
Software Repository Cache34
Software Repository Multimaster Component34
Web Services Data Access Engine34
Opware Gateway34
Global File System Server35
Interaction Among Opware SAS Components35

General Interaction Among Components	35
Opware SAS Security	36
OS Provisioning	36
Patch Management	40
Software Management	48
Code Deployment and Rollback	51
Script Execution	54
Integration with AIX and HP-UX Installation Technology	57
Component Interaction in Multiple Facilities	59
Discovery and Agent Deployment	61
Application Configuration Management	63
Visual Packager	65
Audit and Remediation	67

Chapter 2: User and Group Setup 71

Users, Groups, and Permissions	71
Opware Users and User Groups	71
Opware Permissions	72
Folder Permissions	74
Membership in Multiple Groups	76
Restricted Views of the SAS Web Client	77
Predefined User Groups	78
Special Admin User and Administrators Group	78
Process Overview for Security Administration	79
Managing Users	80

Creating a User80
Editing User Information81
Viewing a User's Permissions81
Deleting a User81
Managing User Groups and Permissions82
Creating a User Group82
Assigning a User to a Group82
Setting the Customer Permissions83
Setting the Facility Permissions83
Setting the Device Group Permissions84
Setting the General Feature Permissions85
Setting the Opware SAS Client Features Permissions86
Setting the Other Features Permissions87
Setting the Permissions for the Opware Global Shell File System87
Setting Folder Permissions88
Delegating Folder Permissions89
Managing the Special Administrators Group89
Adding a User to the Administrators Group89
Removing a User from the Administrators Group90
Password Policy Parameters90
Enabling and Configuring Password Policy Parameters91
Disabling Password Policy Parameters93
External LDAP Directory Service with Opware SAS93

Imported Users	94
SSL and External Authentication.	94
Supported External LDAP Directory Servers	94
Using an LDAP Directory Server with Opsware SAS.	95
Modifying the Web Services Data Access Engine Configuration File . . .	95
Importing a Server Certificate from the LDAP into Opsware SAS	99
Configuring the JAAS Login Module (loginModule.conf)	100
Importing External LDAP Users	101
Code Deployment Permissions	102
Adding Members to a Code Deployment User Group	102
 Chapter 3: Multimaster Mesh Administration	 105
 Overview of Opsware Multimaster Mesh	 105
Multimaster Facilities Administration	106
Updating Facility Information and Settings	106
Multimaster Mesh Administration	108
Overview of Multimaster Mesh Administration	108
Model Repository Multimaster Component Conflicts	109
Causes of Conflicts	110
User Overlap	110
User Duplication of Actions	111
Connectivity Problems that Cause Out of Order Transactions.	111
Best Practices for Preventing Multimaster Conflicts	112
Examining the State of the Multimaster Mesh	113
Best Practices for Resolving Database Conflicts	114

Types of Conflicts.....	114
Guidelines for Resolving Each Type of Conflict.....	115
Model Repository Multimaster Component Conflicts.....	117
Overview of Resolving Model Repository Multimaster Component Conflicts.....	117
Resolving a Conflict by Object.....	118
Resolving a Conflict by Transaction.....	124
Network Administration for Multimaster.....	128
Multimaster Alert Emails.....	128
Chapter 4: Satellite Administration	133
Overview of the Opware Satellite.....	133
Opware Gateway.....	135
Facilities and Realms.....	135
Satellite Information and Access.....	136
Permissions Required for Managing Satellites.....	136
Viewing Facilities.....	137
Enabling the Display of Realm Information.....	139
Viewing the Realm of a Managed Server.....	139
Viewing Gateway Information.....	141
Software Repository Cache Management.....	145

Availability of Packages on the Software Repository Cache	146
Ways to Distribute Packages to Satellites	146
Setting the Update Policy	149
On-demand Updates	150
Manual Updates	150
Hierarchical Software Repository Caches	151
Cache Size Management	151
Creation of Manual Updates	152
Creating a Manual Update Using the DCML Exchange Tool (DET) . . .	152
Applying a Manual Update to a Software Repository Cache	154
Staging Files to a Software Repository Cache	155
Microsoft Utility Uploads and Manual Updates	156
 Chapter 5: Opware SAS Maintenance	 157
 Possible Opware SAS Problems	 157
Opware Component Troubleshooting	158
Contacting Opware Support	158
 Opware SAS Diagnosis	 159

Opware SAS Diagnosis Tool Functionality	159
System Diagnosis Testing Process	160
System Diagnosis Test Components	160
Data Access Engine Tests	161
Software Repository Tests	162
Web Services Data Access Tests	162
Command Engine Tests	163
Model Repository Multimaster Component Tests	163
Running a System Diagnosis of Opware Components	164
The Health Check Monitor for an Opware SAS Core	166
Overview of HCM Local Tests	166
Validating Core Components by Running HCM Local Tests	166
Syntax of the Script for HCM Local Tests	167
Overview of HCM Global Tests	168
Validating a Core By Running HCM Global Tests	169
Syntax of the Script for HCM Global Tests	169
Setting up Passwordless SSH for Global Tests	171
Extensibility of the Health Check Monitor	171
Requirements for Extensions to HCM Local Tests	172
Categories and Local Test Directories	174
Directory Layout for HCM Local Tests	174
HCM Local Test Example	175
Requirements for Extensions to HCM Global Tests	175
HCM Global Test Example	177
Directory Layout for HCM Global Tests:	178
HCM Global Test Directories	178
Logs for Opware Components	178

Boot Server Logs	179
Build Manager Logs	179
Command Engine Logs	179
Data Access Engine Logs	179
Media Server Logs	179
Model Repository Logs	180
Model Repository Multimaster Component Logs	180
Opware Agents Logs	180
SAS Web Client Logs	180
Software Repository Logs	180
Software Repository Replicator Logs	181
Software Repository Multimaster Component Logs	181
Web Services Data Access Engine Logs	181
Opware Gateway Logs	182
Global File System Server Logs	182
Global Shell Audit Logs	182
Shell Event Logs	183
Shell Stream Logs	184
Shell Script Logs	185
Example of Monitoring Global Shell Audit Logs	185
Digital Signatures in the Global Shell Audit Logs	186
Storage Management for the Global Shell Audit Logs	186
Configuring the Global Shell Audit Logs	188
Start Script for Opware SAS	188

Syntax of the Start Script	190
Starting an Opware SAS Core	192
Starting a Multiple-Server Opware SAS Core	192
Starting an Opware SAS Core Component	192
Opware Software	194
Mass Deletion of Backup Files	195
Syntax of Backup Deletion Script	195
Deleting Backup Files with the Mass Deletion Script	196
Designations for Multiple Data Access Engines	198
Overview of Designations for Multiple Data Access Engines	198
Reassigning the Data Access Engine to a Secondary Role	199
Designating the Multimaster Central Data Access Engine	200
Web Services Data Access Engine Configuration Parameters	201
Changing a Web Services Data Access Engine Parameter	201
Web Services Data Access Engine Configuration File	201
Opware Global File System Administration	204
Additional Option Needed for passwd Command on OGFS Server	204
Chapter 6: Monitoring Opware SAS	205
Overview of Opware SAS Monitoring	206
Opware Agent Monitoring	207
Agent Cache Monitoring	210
Opware Command Center Monitoring	211
Load Balancing Gateway Monitoring	212
Data Access Engine Monitoring	213
Web Services Data Access Engine Monitoring	215
Command Engine Monitoring	218

Software Repository Monitoring219
Model Repository Monitoring224
Model Repository Multimaster Component Monitoring226
TIBCO Monitoring227
Opsware Global File System Server Monitoring230
Spoke Monitoring233
Opsware Gateway Monitoring234
OS Build Manager Monitoring236
OS Boot Server Monitoring238
OS Media Server Monitoring238
Chapter 7: Opsware SAS Configuration	241
<hr/>	
System Configuration241
Ways to Use Opsware SAS Configuration Parameters241
Configuring Contact Information in the Opsware Help	242
Configuring the Mail Server for a Facility	243
Setting Email Alert Addresses for an Opsware Core	244
Configuring Email Alert Addresses for Multimaster	245
Configuring Email Notification Addresses for CDR	245
Appendix A: Permissions Reference	249
<hr/>	
Permissions Required for the SAS Web Client249
Permissions Required for the Opsware SAS Client251

More Information for Security Administrators	251
Application Configuration Management Permissions	252
Device Group Permissions	260
ODAD Permissions	261
Job Permissions	261
Patch Management for Windows Permissions	262
Patch Management for Unix Permissions	267
Software Management Permissions	270
Audit and Remediation Permissions	282
Visual Application Manager Permissions	293
Virtualization Director Permissions	294
Visual Packager Permissions	296
OS Provisioning Permissions	298
Script Execution Permissions	301
Predefined User Group Permissions	303
Code Deployment User Groups	308
Appendix B: Software Repository Replicator Setup	311
Overview of the Software Repository Replicator	311
Prerequisites for Using the Software Repository Replicator	311
Software Repository Replicator Configuration	312
Sample Software Repository Replicator Configuration	313
Index	315

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 create and administer Opsware SAS user accounts, and how to administer multimaster facilities and Opsware Satellites. 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 in single and multiple core environments. Each of the components and its function is introduced.

Chapter 2: User and Group Setup: Provides information about how to create and delete users, user groups, and administrators and how to assign permissions to each.

Chapter 3: 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 4: Opsware Satellite Administration: Provides overview information about an Opsware Satellite facility and how to administer one after installation.

Chapter 5: 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.

Chapter 6: Monitoring Opsware SAS: Provides information about performing basic monitoring of the Opsware SAS components in a core so that you can automate Opsware SAS system diagnosis and monitoring. The type of monitoring information described in the chapter includes the commands to confirm specific component processes are running, as well as examples of the expected output, and component specific ports, logs, and administrative URLs.

Chapter 7: Opsware SAS Configuration: Provides information about how to set several configuration parameter values that Opsware SAS uses to send email notifications and alerts, and to display the Opsware administrator contact information.

Appendix A: Permissions Reference: Provides information about which Opsware SAS permissions to grant Opsware users so that they access only the areas of functionality relevant to their responsibilities in the managed server environment.

Appendix B: Software Repository Replicator Setup: Describes how to set up the Software Repository Replicator to enable backup functionality for Software Repositories running in a multimaster mesh.

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.
Courier	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.

NOTATION	DESCRIPTION
<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 icons.

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 *Opware[®] SAS User's Guide: Server Automation* is intended for system administrators responsible for all aspects of managing servers in an operational environment. It describes how to use Opware SAS, introducing the system and the user interface. It provides information about managing servers, remediating servers,

script execution, configuration tracking, deploying and rolling back code, and agent deployment. It also explains how to use the Opsware Global Shell and open a Remote Terminal on managed servers.

- *Opsware® SAS User's Guide: Server Automation* is intended for system administrators responsible for performing the day-to-day functions of managing servers. It reviews auditing and compliance, software packaging, visual application management, application configuration, and software and operating system installation on managed servers.
- The *Opsware® SAS Administration Guide* is intended for administrators responsible for monitoring and diagnosing the health of the Opsware SAS core components. It also documents how to set up Opsware user groups and permissions.
- The *Opsware® SAS Planning and Installation Guide* is intended for advanced system administrators responsible for planning all facets of an Opsware SAS installation. It documents all the main features of Opsware SAS, scopes out the planning tasks necessary to successfully install Opsware SAS, explains how to run the Opsware Installer, and details how to configure each of the components. It also includes information on system sizing and checklists for installation.
- The *Opsware® SAS Policy Setter's Guide* is intended for system administrators responsible for setting up OS provisioning, configuration tracking, code deployment, and software management.
- The *Opsware® SAS Content Utilities Guide* is intended for advanced system administrators responsible for importing content such as software packages into Opsware SAS. It documents the following command-line utilities: OCLI 1.0, IDK, and DET (CBT).
- The *Opsware® Automation Platform Developer's Guide* is intended for software developers responsible for customizing, extending, and integrating Opsware SAS. It documents how to create Web Services, Java RMI, Python, and CLI clients that invoke methods on the Opsware API.

Opsware, Inc. Contact Information

For more information, see the Opsware, Inc. main web site and phone number:

- <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, see the Opsware Knowledge Base:

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

To contact Opsware Customer Support, see the following email address and phone number:

- 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.

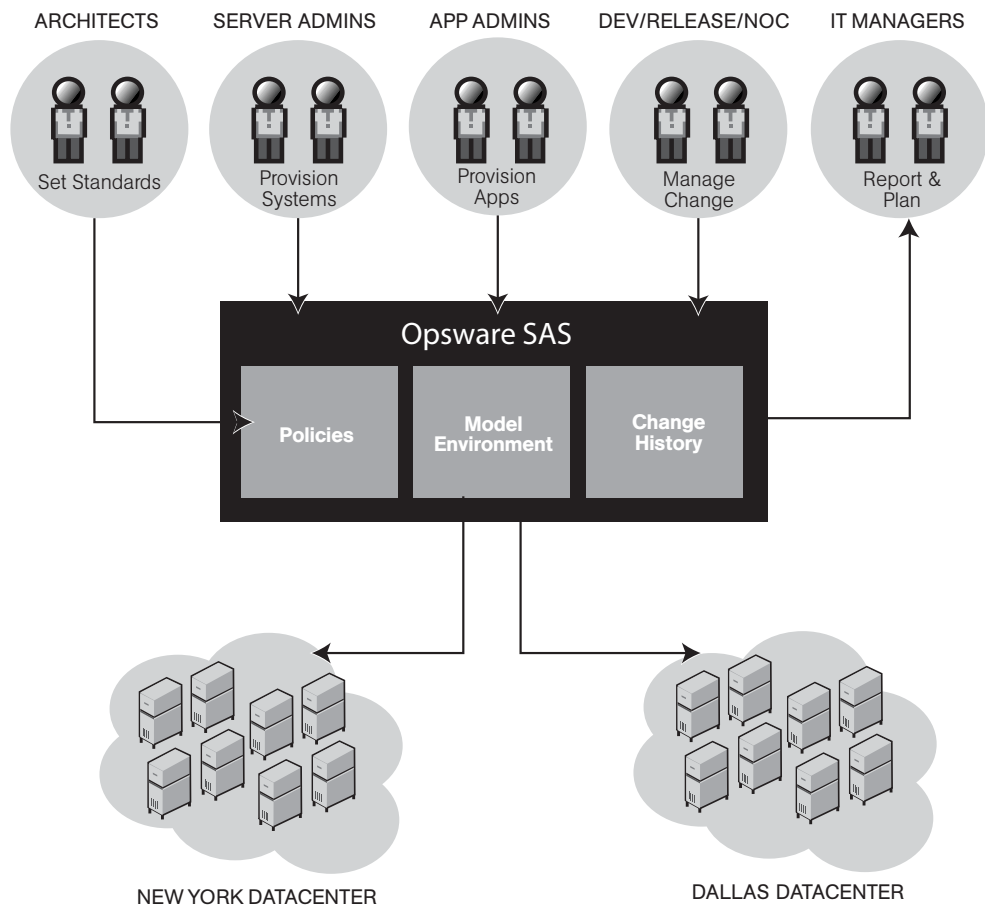
Opsware SAS does not just automate your operations, it also allows you to make changes more safely and consistently, because you can model and validate changes before you actually commit the changes to a server. Opsware SAS helps ensure that modifications to your servers work on your first attempt, thereby 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 Opware 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 Opware 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 management policies.
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 SAS Web Client and SAS 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 SAS Web Client 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 Policies, preparing Operating System Definitions, and acting as Patch Administrators to approve patches for installation in the operational environment.

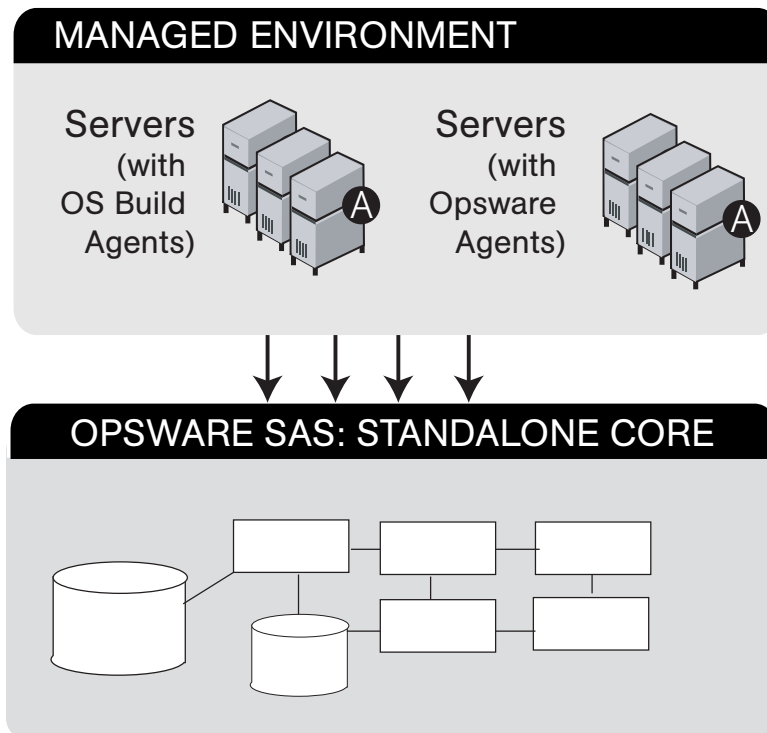
Opware SAS Environment

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 SAS Web Client, 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.

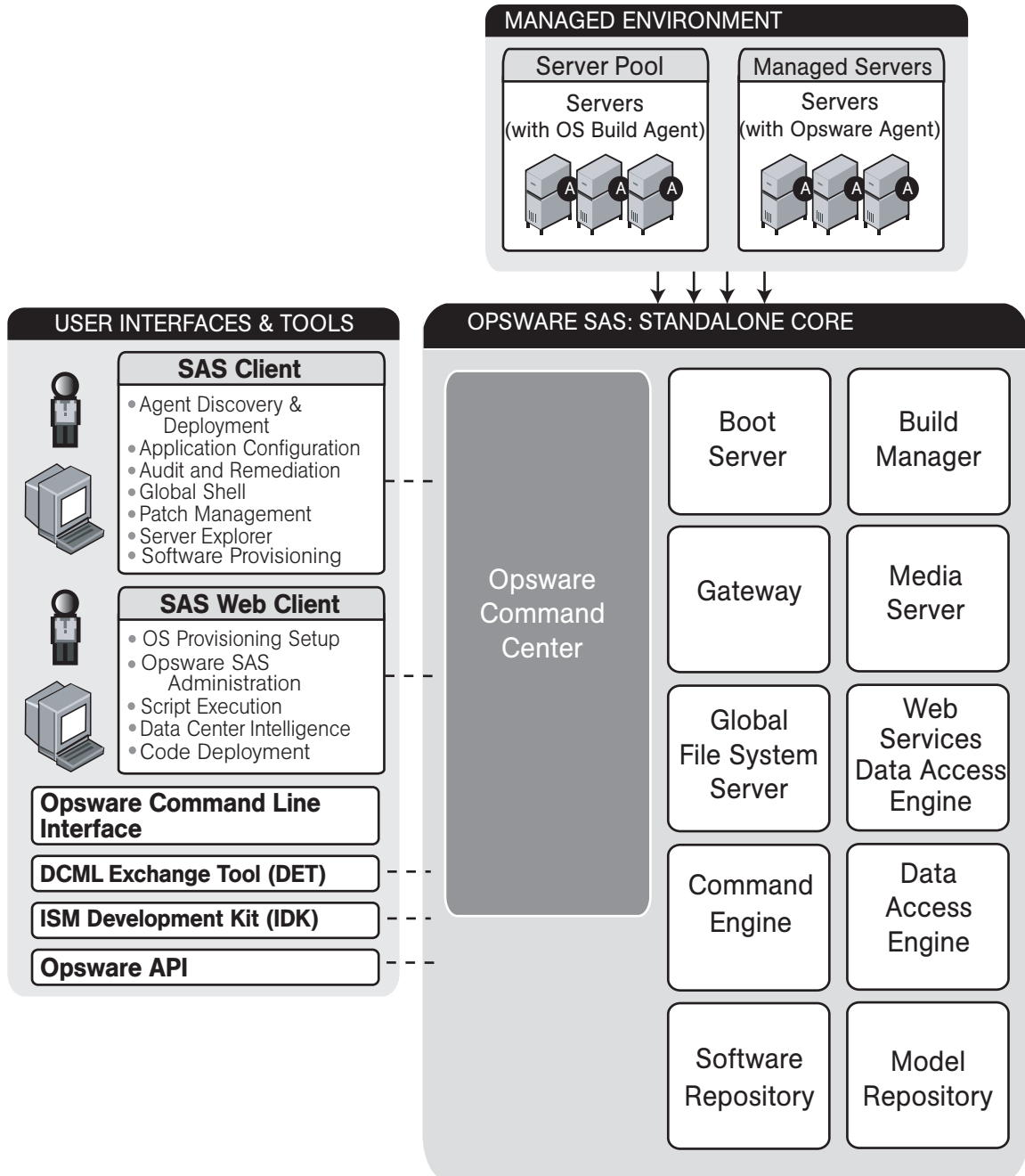
The sections that follow describe the components of Opsware SAS:

- **Boot Server:** The 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.
- **SAS Client:** The Windows user interface to Opsware SAS.
- **SAS Web Client:** The browser-based interface to Opsware SAS.
- **Opsware Command Center:** The core component that communicates with the SAS Web Client
- **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 hosts Global Shell sessions 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 Opsware 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 SAS Web Client, 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 Opware Agent might use global Opware SAS services (such as the Model Repository and Software Repository) in order to fulfill the request.

Some functions that the Opware Agent supports are:

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

An Opware Agent is idle unless Opware SAS is trying to perform some change on the server. Each Opware 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 Opware Agents

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

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

Opware SAS Client

The Opware SAS Client is a Windows user interface to Opware SAS. With the Opware SAS Client, the user can access most of the Opware SAS features, including Software Management, Patch Management, Audit and Remediation, Application Configuration, and the Server Explorer.

Opware SAS Web Client

The Opware SAS Web Client is a browser-based user interface to Opware SAS. Through this interface, an Opware SAS user can use the OS Provisioning and Code Deployment and Rollback (CDR) features. An Opware administrator manages users and defines security permissions with this interface.

Opware Command Center

The core component that communicates with the Opware SAS Web Client. The Opware Command Center (OCC) includes an HTTPS proxy server and an application server. The OCC is installed on the same server as the Web Services Data Access Engine.

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 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 remediation.

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 Policy Setter's 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.

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 and provides increased performance to other Opware SAS components. This object abstraction can be accessed through a Simple Object Access Protocol (SOAP) API, by third-party integration components, or by a binary protocol of Opware SAS components like the SAS Web Client.

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 serve 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, a core component that dynamically constructs a virtual file system – the Opsware Global File System (OGFS). The Global File System Server can connect to an Opsware Agent to open a Unix shell or to open 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 Management
- 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
- Audit and Remediation

General Interaction Among Components

The SAS Web Client, 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 Microsoft WIM imaging is supported. Image-based provisioning using Symantec Ghost and Sun Solaris Flash is supported, but not 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 Software Management 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: Application Automation* for information about the OS provisioning process.

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

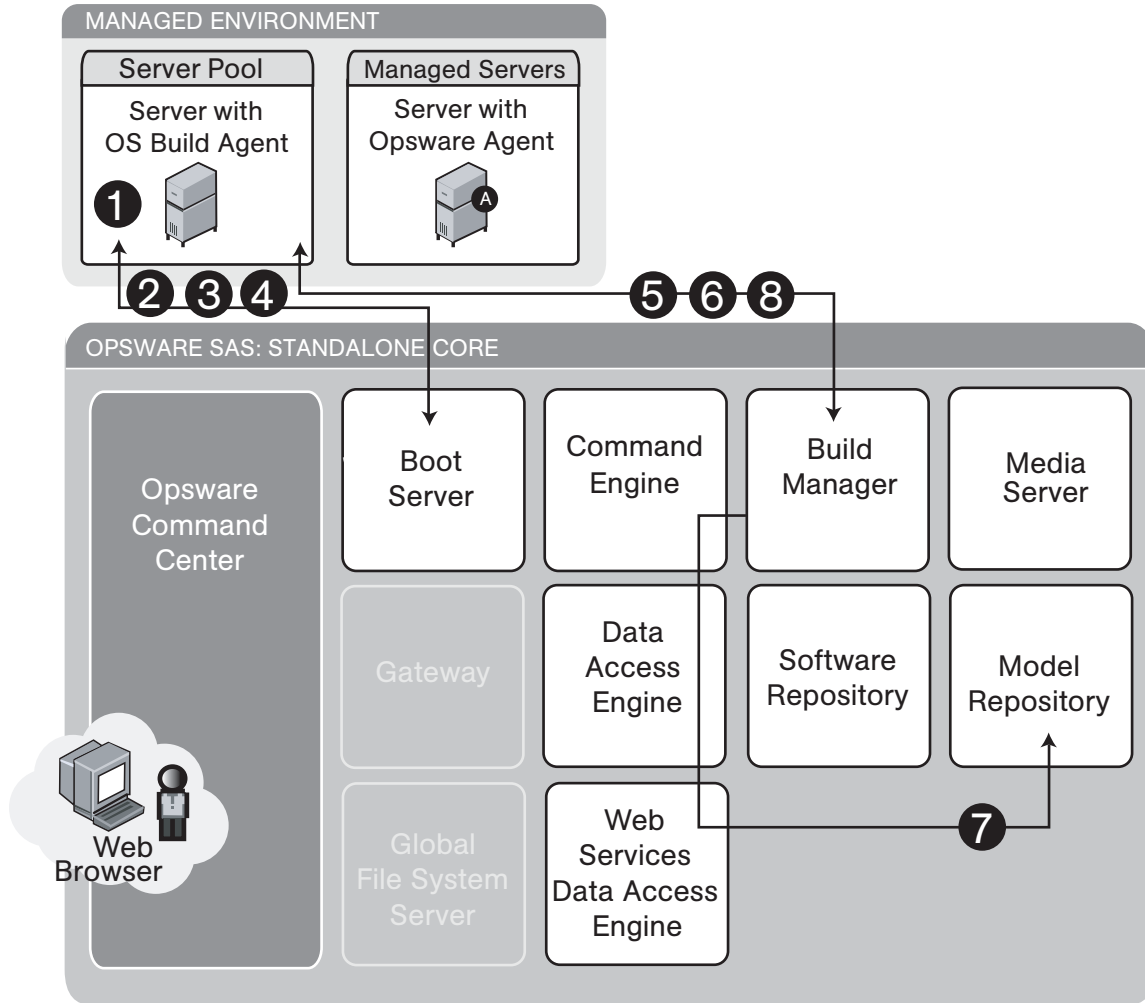


Figure 1-5: OS Provisioning Step 2: OS Installation

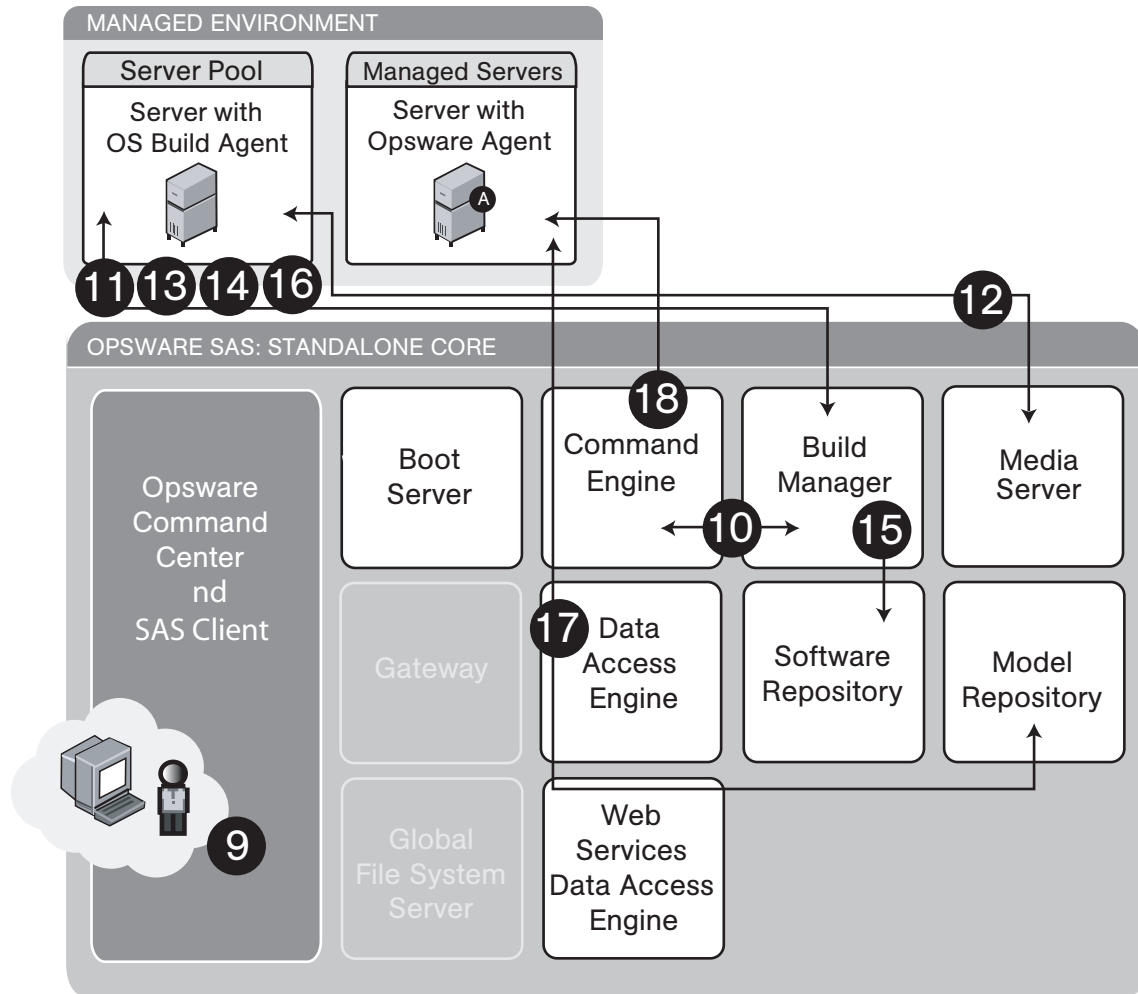


Figure 1-4 and Figure 1-5 illustrate the OS provisioning process:

OS Provisioning Step 1: Initial Booting:

- 1** The DHCP request is a network broadcast.
- 2** The DHCP reply contains the IP address and TCP port of the appropriate Opware core agent gateway for use by Solaris JumpStart and WinPE-based Windows provisioning. For Microsoft Windows provisioning using the DOS-based preinstallation environment, the gateway IP and port combination is embedded in the floppy images by the Opware installer at install-time.

- 3** TFTP is used to boot the server over the network (using PROM-based inetboot for Solaris SPARC, and PXE for Solaris x86, Windows, Linux, and VMware ESX). Instead of PXE, the DOS-based pre-installation environment for Windows can use a boot floppy and the WinPE pre-installation environment for Windows 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 SAS Web Client or until the server is removed from the network.

OS Provisioning Step 2: OS Installation:

- 9** A user initiates OS provisioning with the Install OS Wizard in the SAS Web Client or runs an OS sequence from the SAS Client. For information on using an OS sequence see *Opsware® SAS User's Guide: Application Automation*.
- 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 SAS Client.
- 11** A Media Resource Locator (MRL) contains the network location (host name and path) of an NFS or SMB server from which to retrieve the vendor OS installation media or obtain the WIM install image.
- 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, VMware ESX or Windows unattended install).
- 14** The server is rebooted after OS installation (depending on the installation type, there might be multiple reboots).
- 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 remediate 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: Application Automation* for information about the patch management process.

Figure 1-6: Patch Management Feature: Import Patches

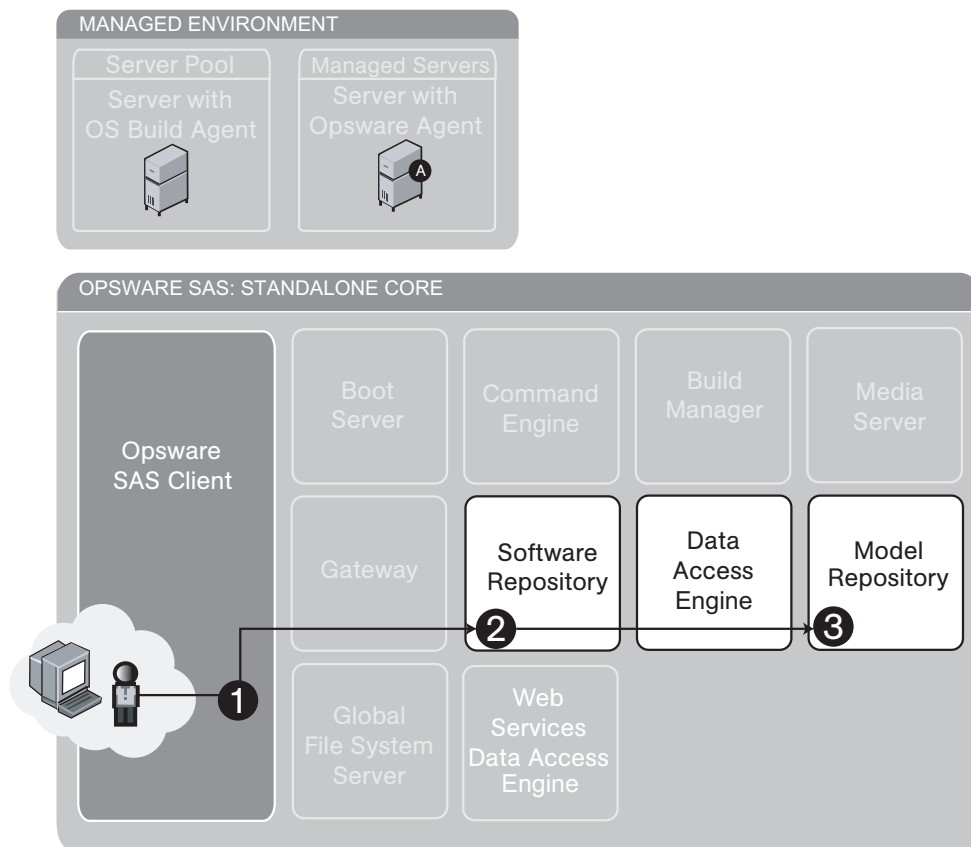


Figure 1-6 illustrates the following import processes for Windows and Unix patches:

Windows Patches

- 1 An Opware user with the required permissions logs into the Opware SAS Client and selects Opware Administration ► Patch Settings from the Navigation pane. To import the patch database from the Microsoft web site, click **Import from Vendor**.

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

See the *Opware® SAS User's Guide: Application Automation* for information about importing the Microsoft patch database.

Unix Patches

- 1 From the Navigation pane, select Library ► By Folder ► Patches.
- 2 An Opware user with the required permissions logs in to the Opware SAS Client and selects **Import Software** from the **Actions** menu. The Import Software window displays.
- 3 Using the Import Software window, the user specifies a Patch type and Platform and uploads the file to the Software Repository.
- 4 The Software Repository places a record of the location, file size, and patch state of each patch in the Model Repository with the Data Access Engine.

See the *Opware® SAS Policy Setter's Guide* for information about the importing software process.

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

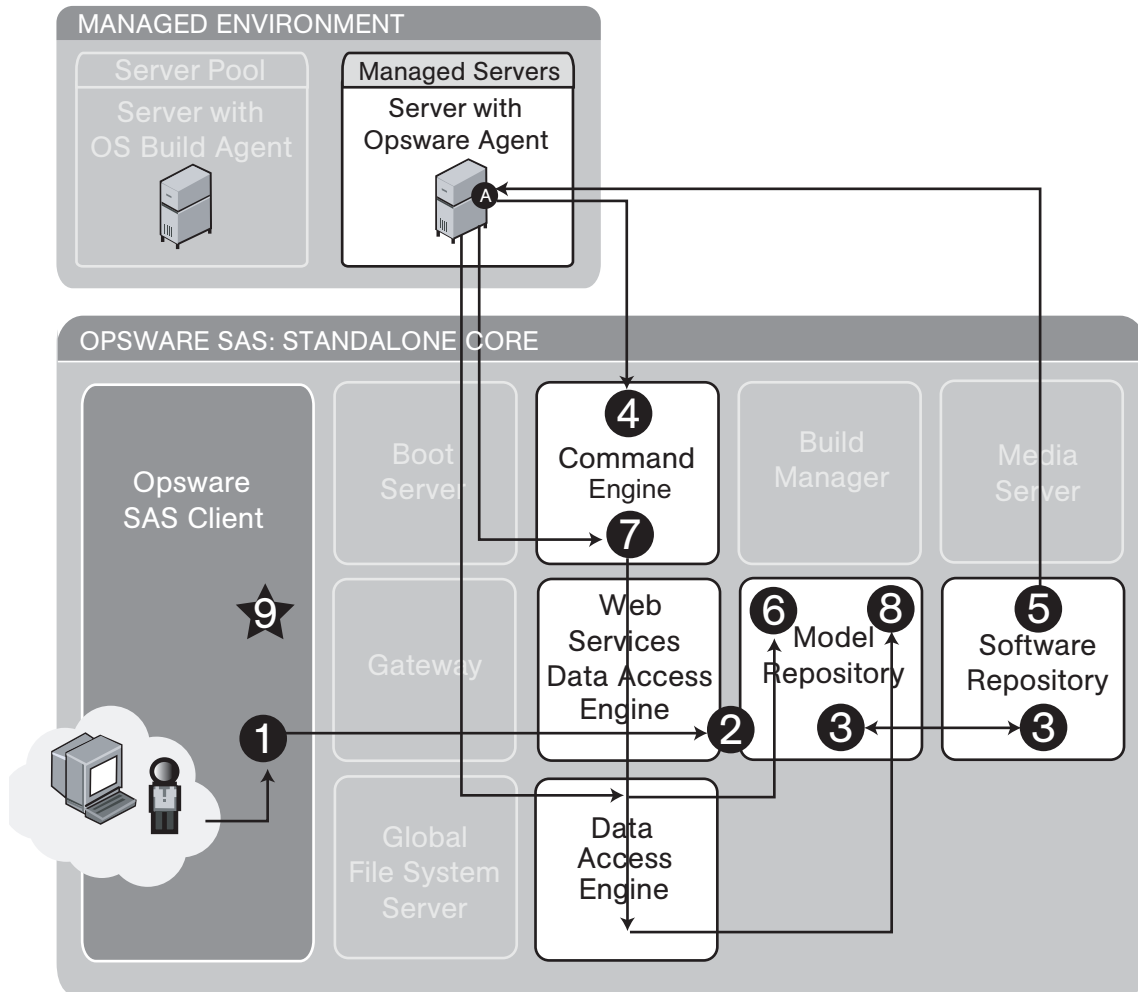


Figure 1-7 illustrates the install patch process:

- 1** An Opware user with the required permissions logs in to the Opware SAS Client and selects **Install Patch** from the **Actions** menu. The Install Patch window displays.
- 2** Using the Install Patch window, the user specifies patches, servers, reboot options, pre and post install scripts, scheduling information, and starts the install process, retrieving patch information from the Model Repository with 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 with 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 with 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 with the Data Access Engine.
- 9** An operation complete status message displays in the Opware SAS Client.

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

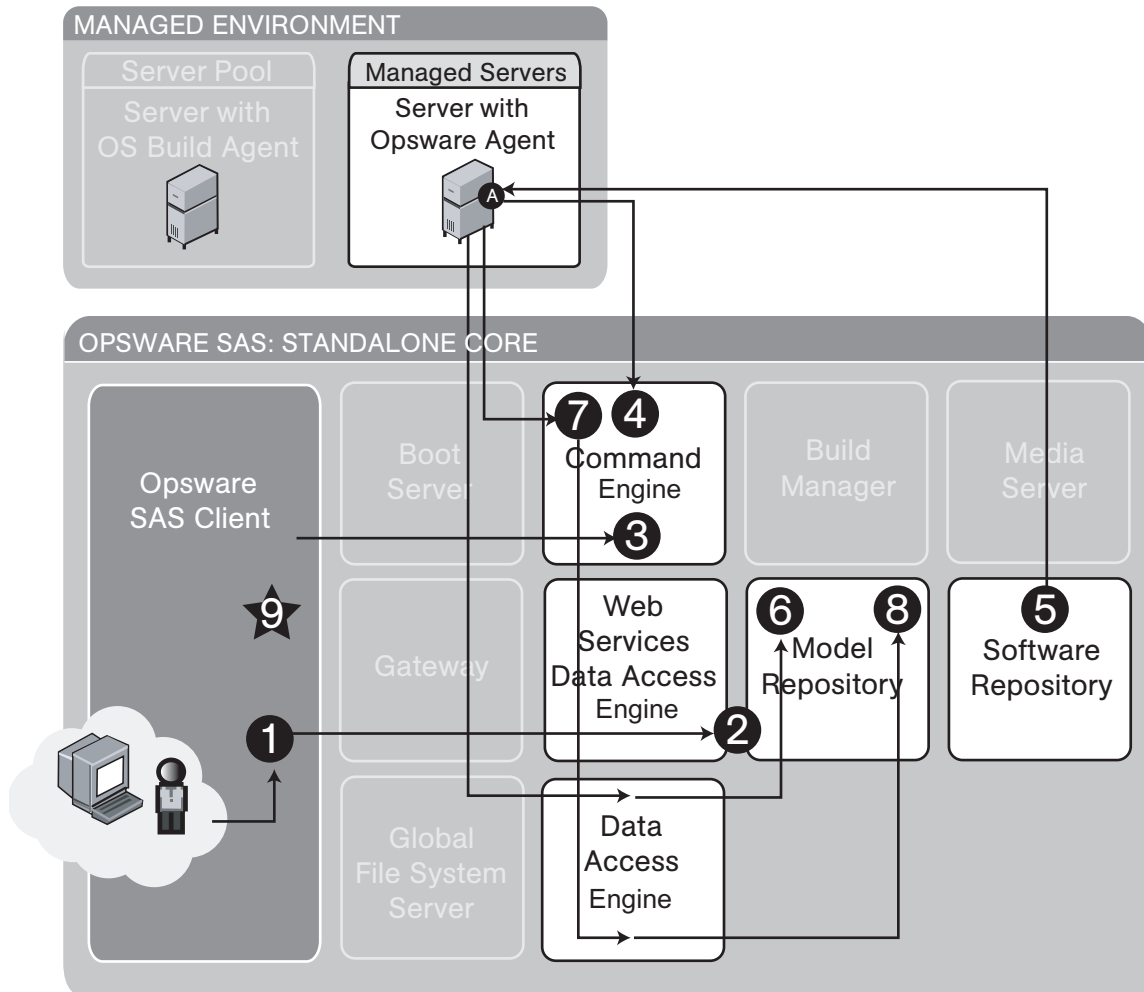


Figure 1-8 illustrates the uninstall patch process:

- 1** An Opsware user with the required permissions logs in to the Opware SAS Client and selects **Uninstall Patch** from the **Actions** menu. The Uninstall Patch window displays.
- 2** Using the Uninstall Patch window, the user specifies patches, servers, reboot options, pre and post uninstall scripts, scheduling information, and starts the uninstall process, retrieving server and patch information from the Model Repository with the Web Services Data Access Engine.
- 3** The Opware SAS Client 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 with 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 with the Data Access Engine.
- 9** An operation complete status message displays in the Opware SAS Client.

Figure 1-9: Patch Management Feature: Patch Policy Remediation Process

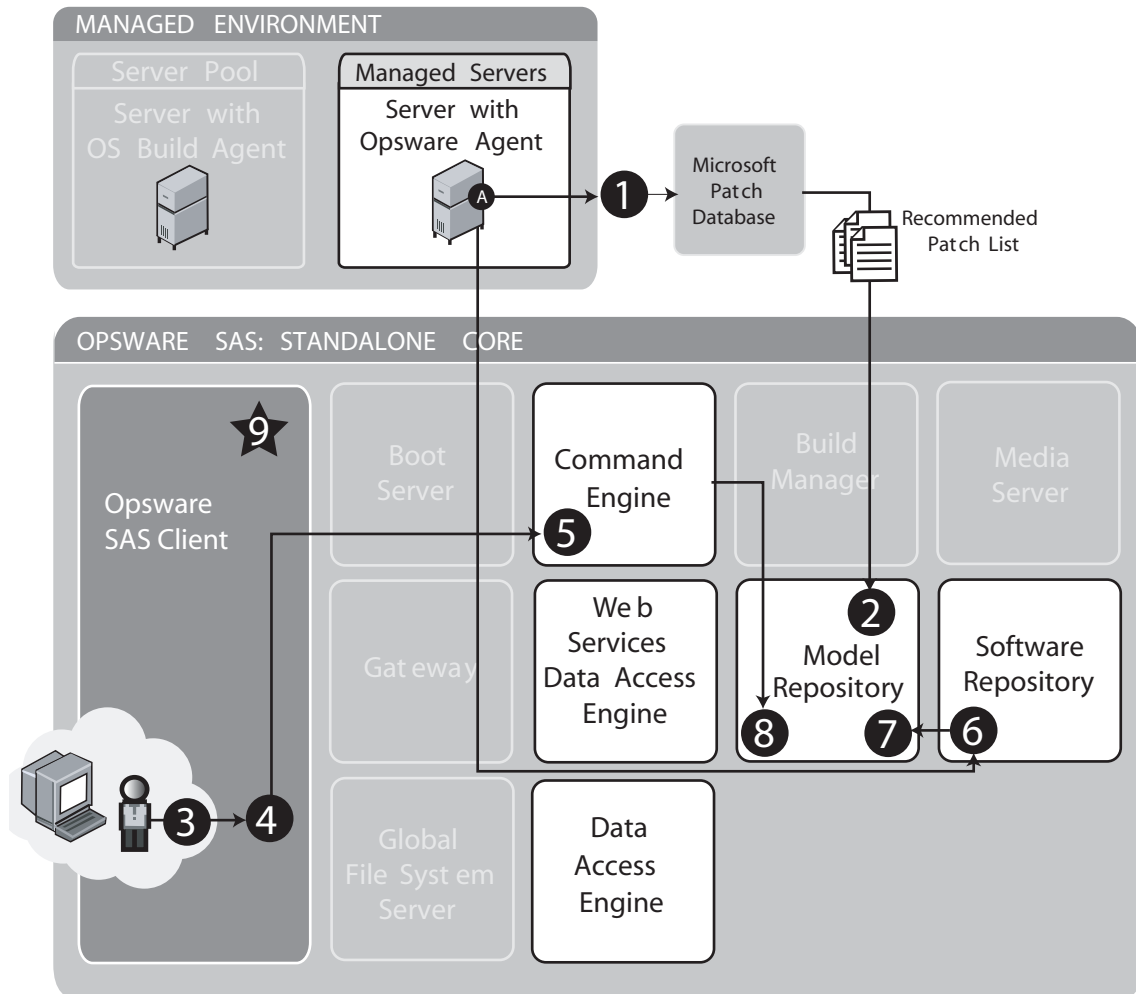


Figure 1-9 illustrates the patch policy remediation process for vendor-recommended (Windows) patches:

- 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 with the Data Access Engine.

- 3** An Opware user with the required permissions logs in to the Opware SAS Client and attaches the vendor-recommended patch policy to the server.
- 4** Using the Remediate window, the user performs the patch policy remediation process to install the patches in the vendor-recommended patch policy.
- 5** The installation details are passed from the SAS 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 with the Data Access Engine.
- 8** Install operation status is reported to the Command Engine, which places it in the Model Repository with the Data Access Engine.
- 9** An operation complete status message displays in the Opware SAS Client.

Software Management

In Opware SAS, packages reside in a central Software Repository. Opware policy setters upload the packages and patches and also specify options that help ensure that the software is installed in a safe and consistent way. Policy Setters then create software policies and add the software resources such as packages, patches, application configurations, and other software policies to the software policy. In a software policy they specify the installation order for software installation. A system administrator then attaches the software policy to a server and remediates the server. During remediation, the software specified in the software policy is installed on the server.

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

Figure 1-10: Software Management Step 1: Preview Remediation

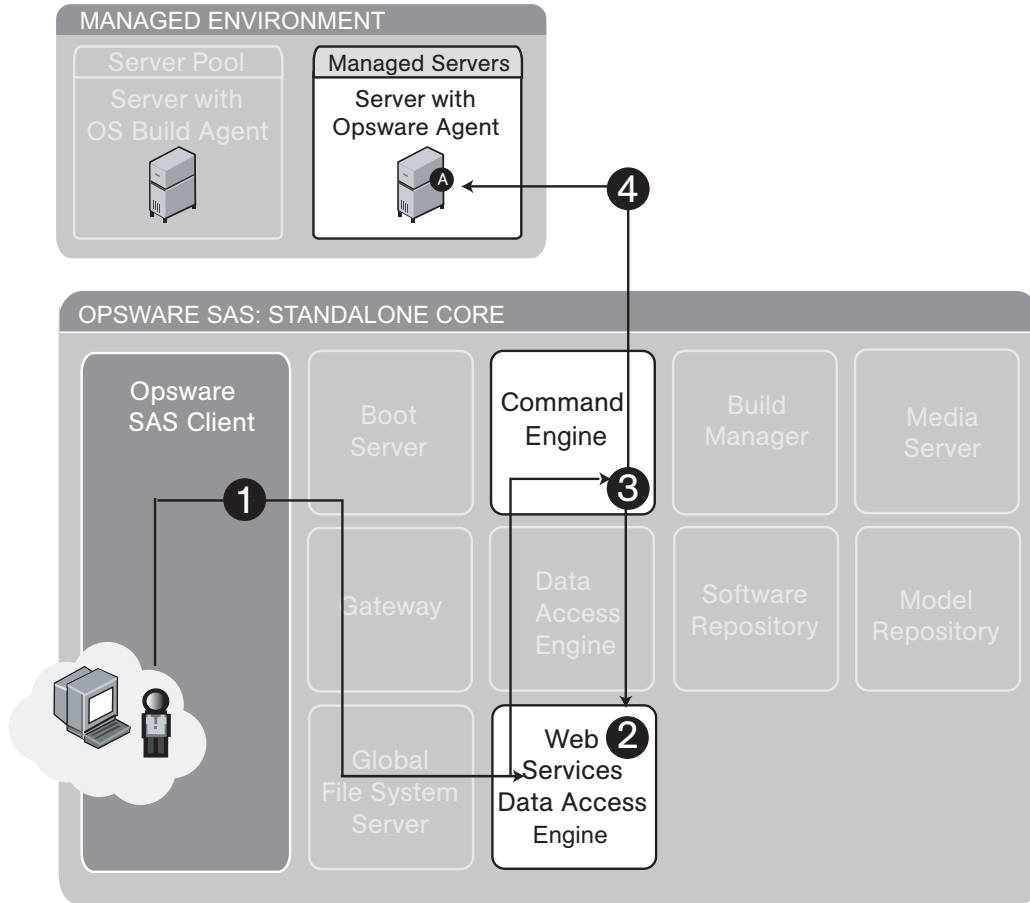


Figure 1-11: Software Management Step 2: Software Installation and/or Removal Through Remediate

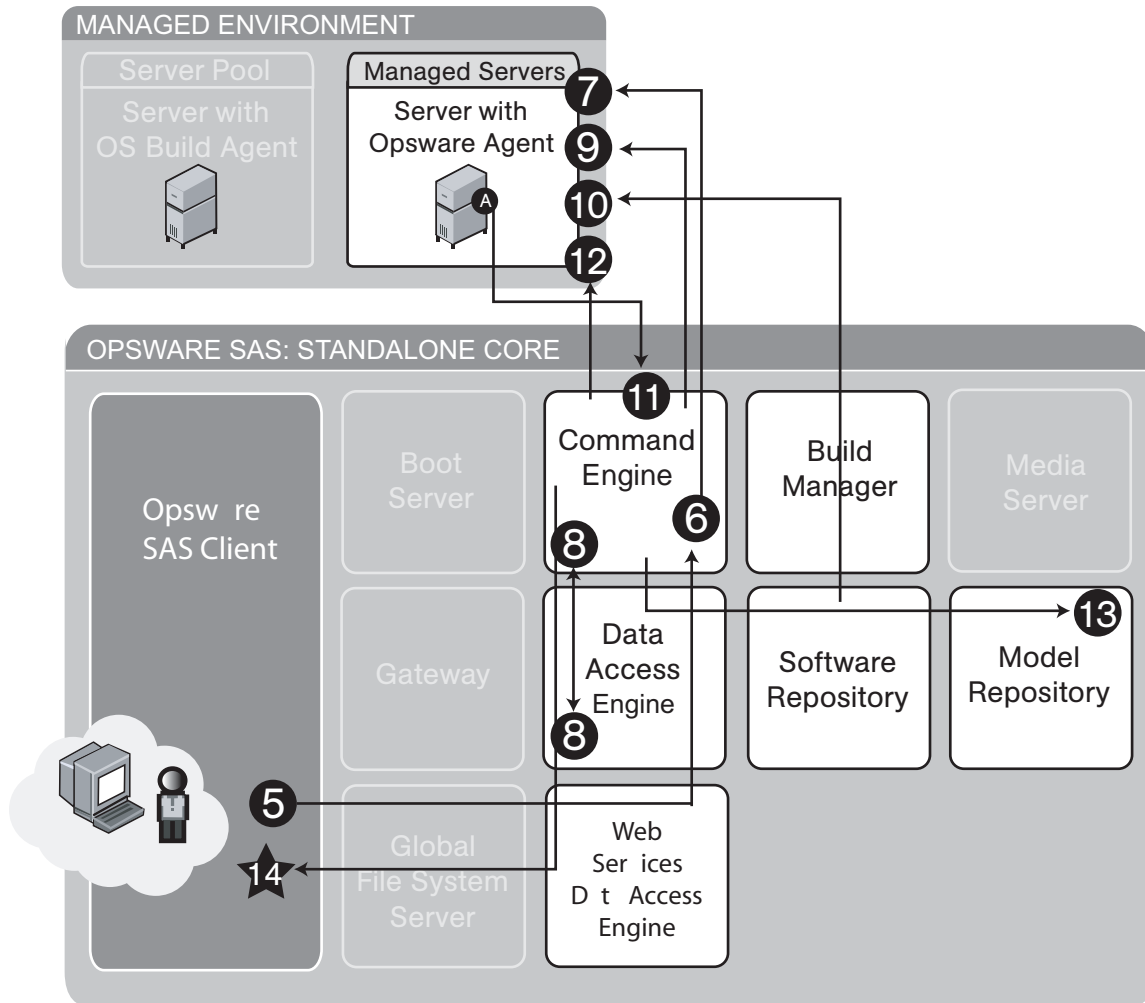


Figure 1-10 and Figure 1-11 illustrate the software installation process:

Software Installation Step 1: Determine Server Configuration:

- 1** An Opware user logs into the SAS Client and selects one or more servers and software policies to remediate against.
- 2** The SAS Client starts the Preview Remediate Job in the Web Services Data Access Engine.
- 3** The Opware Agent on each of the specified servers is queried by the Command Engine for a list of software installed on its server.

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

Software Installation Step 2: Software Installation through Remediate

- 5 At the end of Remediate Preview, the SAS Client displays a list of the software to be installed and/or removed. The user confirms proceeding with the remediate.
- 6 The SAS Client starts the Remediate job in the Command Engine.
- 7 The Opware Agent on each of the specified servers is queried by the Command Engine for a list of software installed on its server.
- 8 The Command Engine, via the Data Access Engine, compares that list to the user-specified list of software policies to determine what needs to be installed or removed.
- 9 The Command Engine tells the Opware Agent to install and/or remove software.
- 10 The Opware Agent downloads software from the Software Repository, removes any software that need to be removed, and installs the new software, performing all necessary install, uninstall, and reboots, if required.
- 11 The Opware Agent reports installation status to the Command Engine.
- 12 The Opware Agent on each of the specified servers is queried by the Command Engine for a list of software 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 software displays in the SAS Client via the Command Engine and the Web Services Data Access 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 Opware SAS staging environment. You can use Opware 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 SAS Web Client navigation panel. Figure 1-12 shows the code deployment and rollback process.

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

Figure 1-12: Code Deployment and Rollback Feature

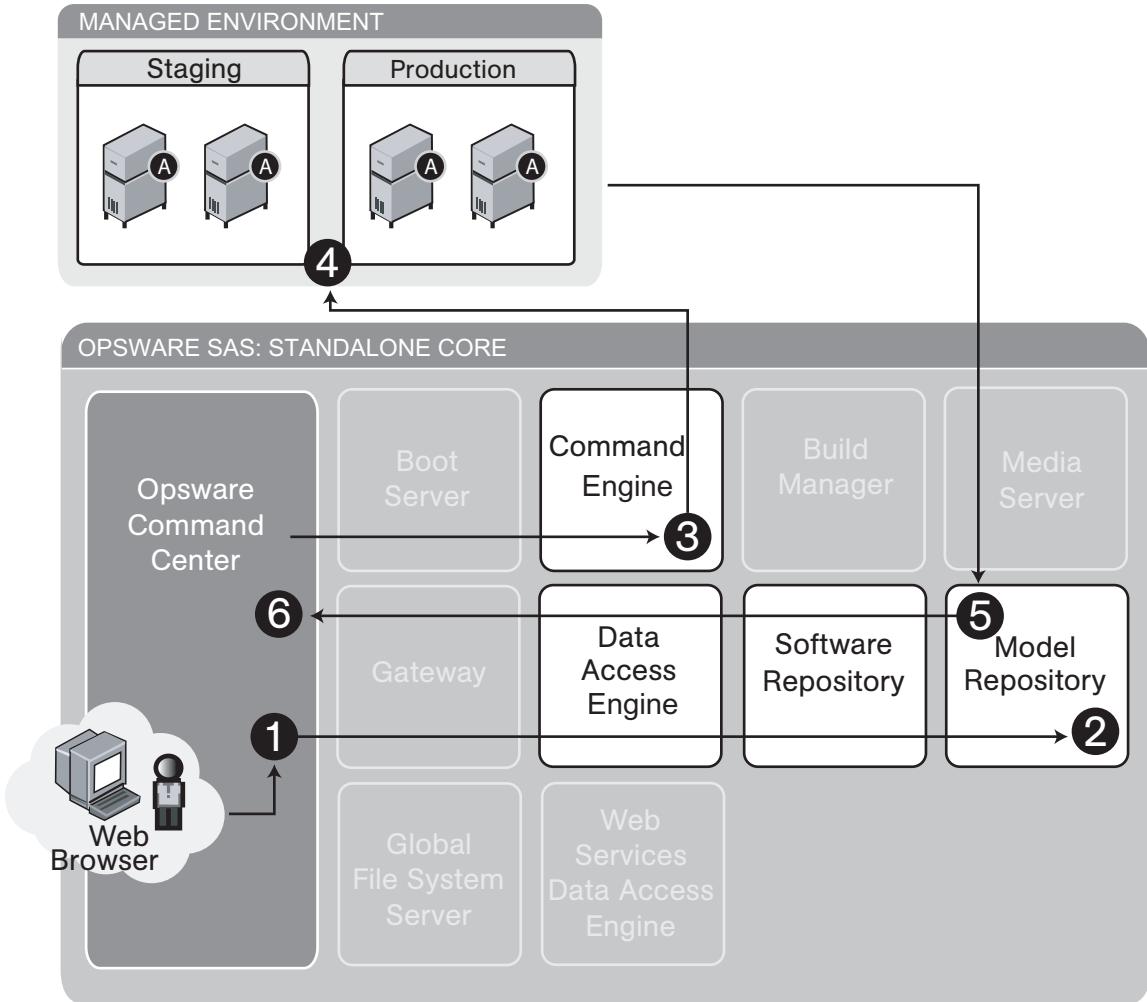


Figure 1-12 illustrates the code deployment and rollback process:

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

From the SAS Web Client, 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: Server Automation* for information about the process to create and execute scripts in the managed environment.

Figure 1-13: Scripting Feature: Upload Script

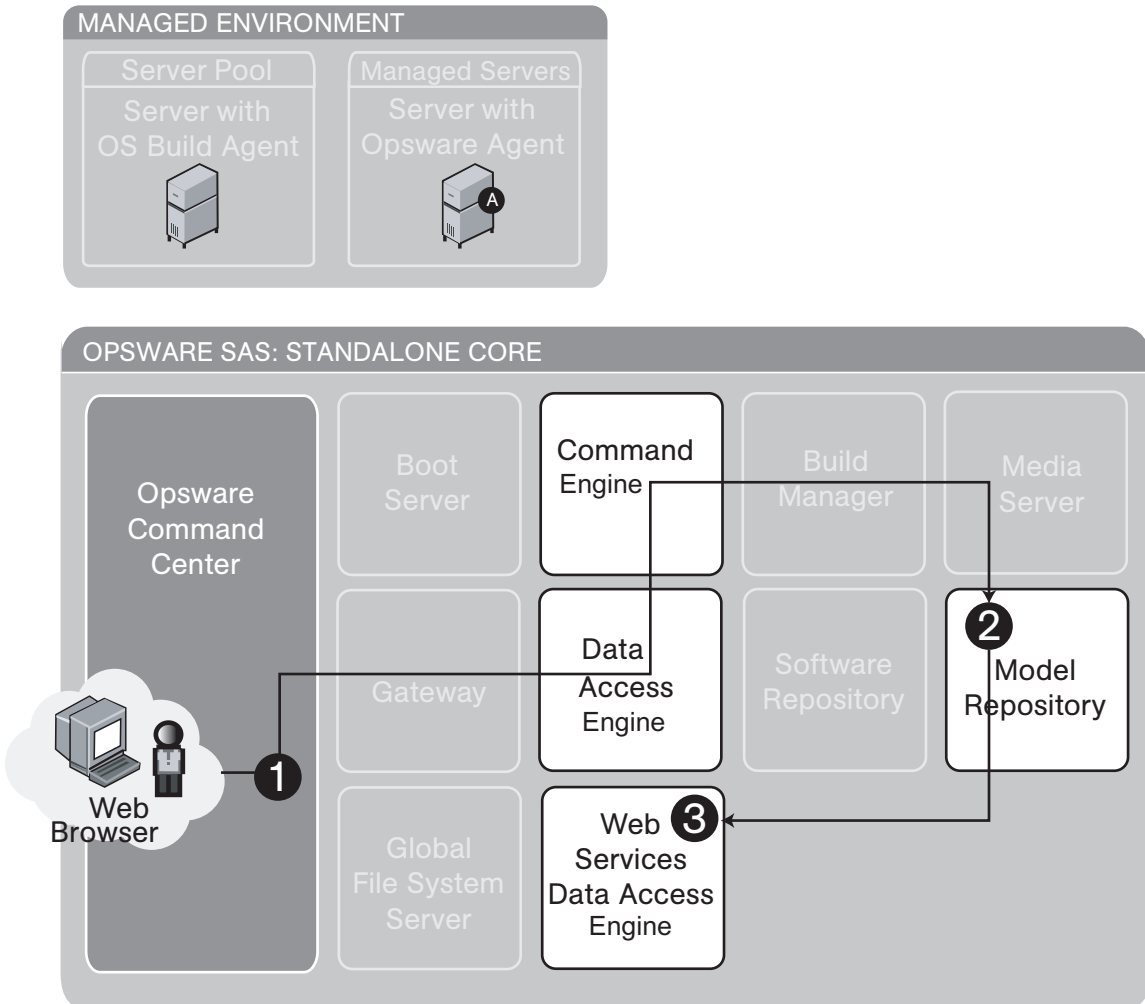


Figure 1-13 illustrates the script execution – upload script process:

- 1** An Opware user with the required permissions logs into the SAS Web Client 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

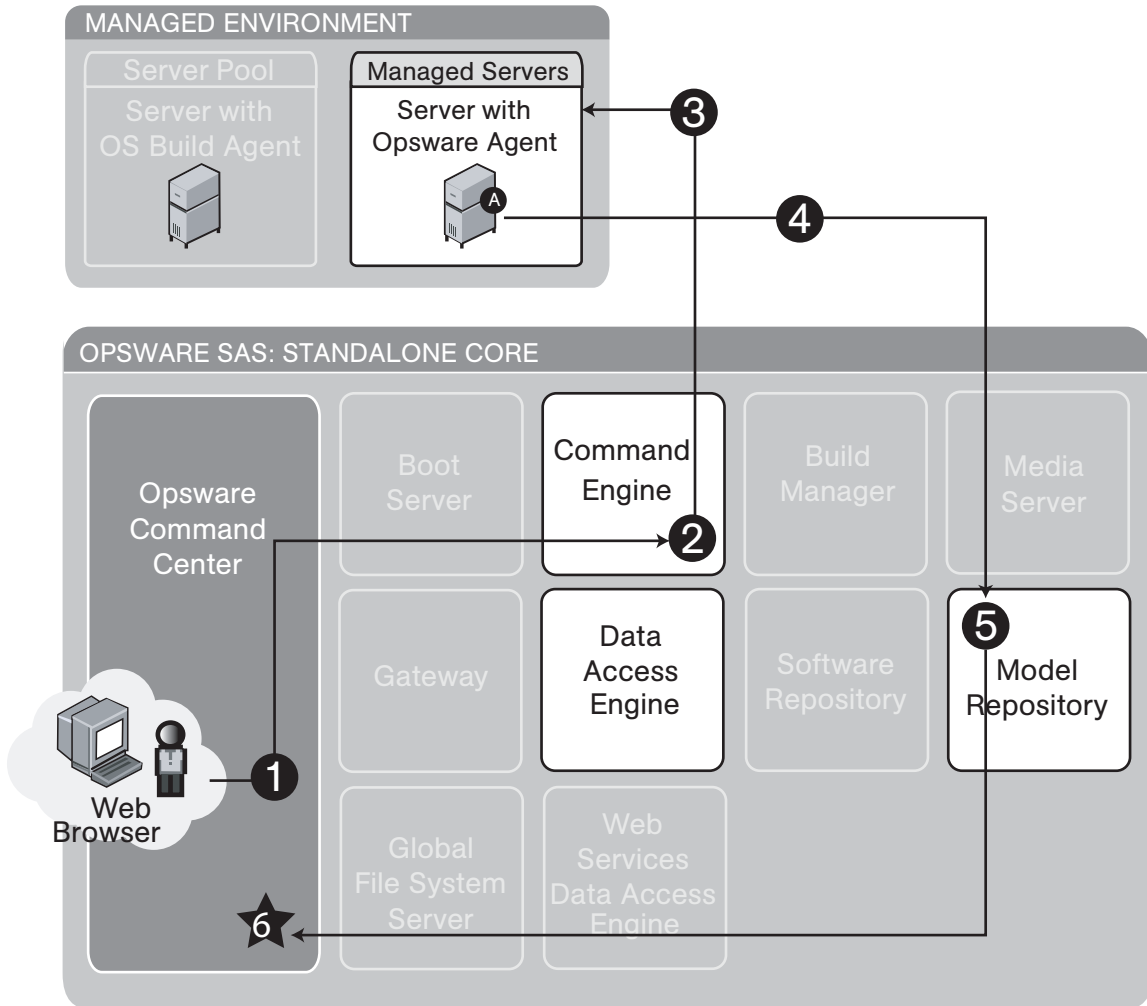


Figure 1-14 illustrates the scrip execution – execute script process:

- 1 An Opware user with the required permissions logs into the SAS Web Client 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 SAS Web Client 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.

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.

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

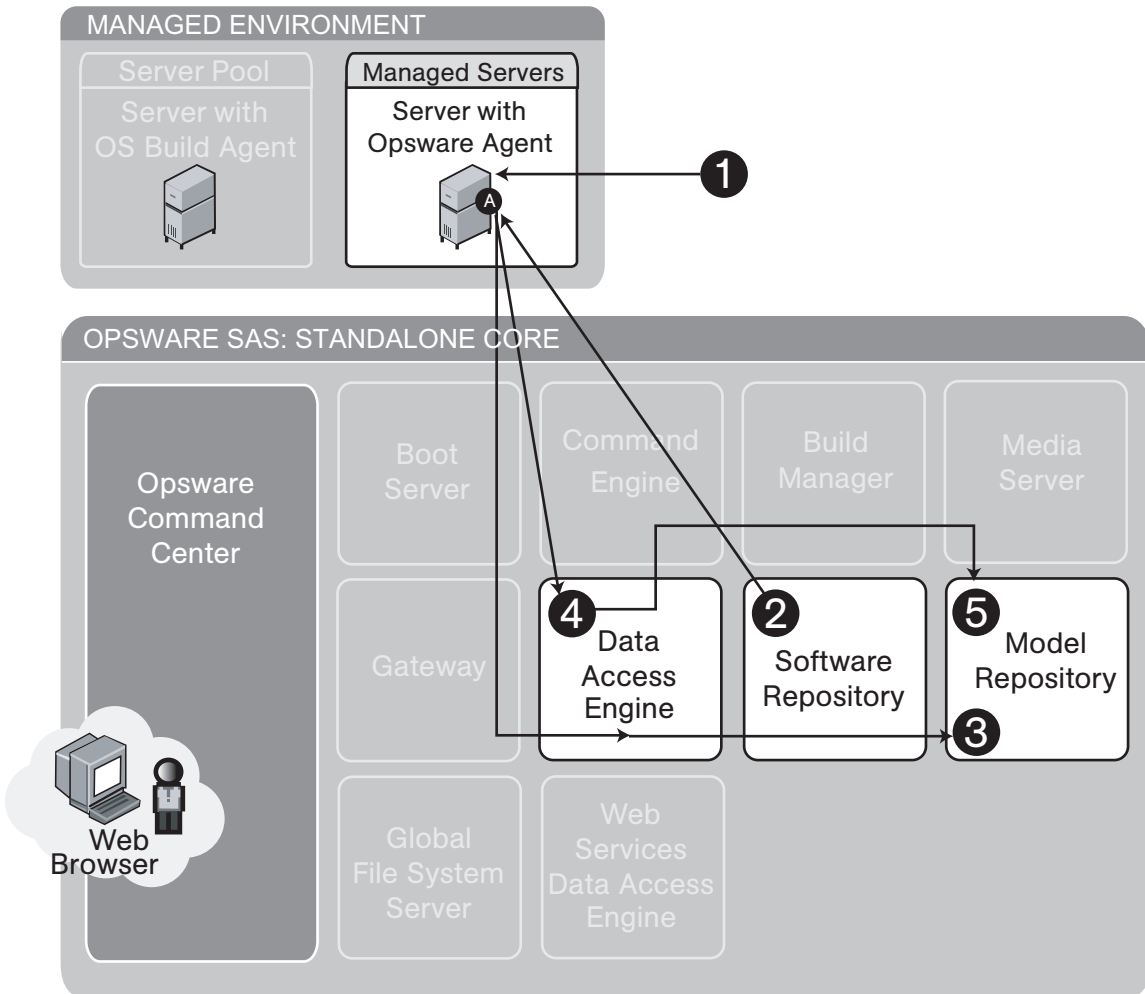


Figure 1-15 illustrates Opware SAS integration with AIX and HP-UX operating systems:

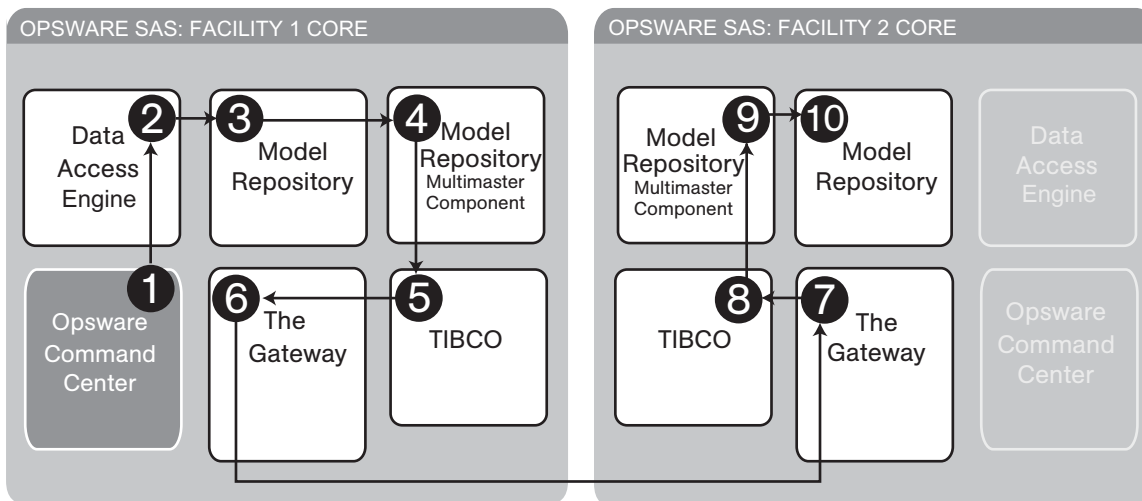
- 1** Installation technology installs the OS.
- 2** Opware SAS integration downloads and installs the Opware Agent on the server.

- 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 remediated 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 108 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

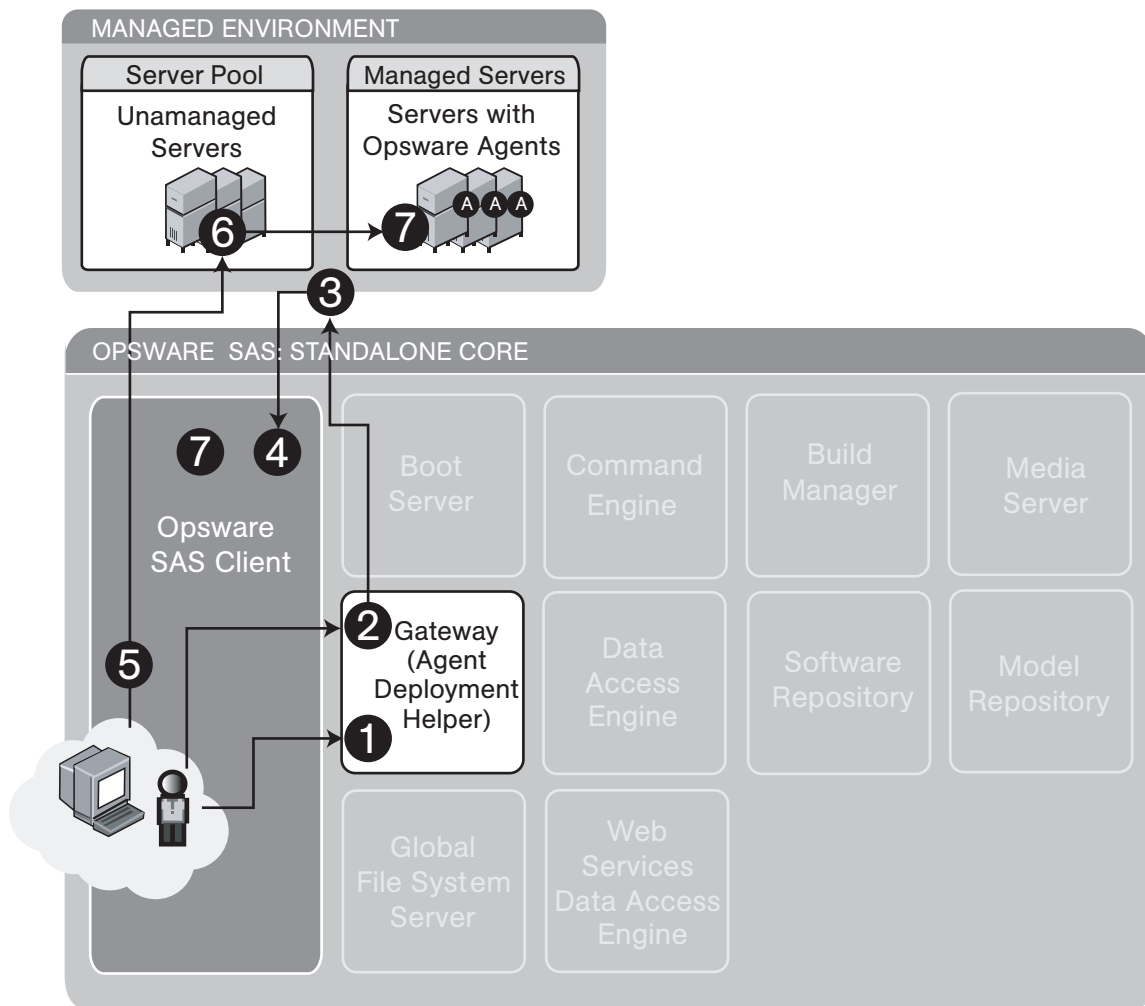


Figure 1-17 shows the process of discovering unmanaged servers and installing the Opware Agent on those servers:

- 1** An Opware user launches the Discover and Agent Deployment feature in the SAS 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 SAS 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 SAS 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

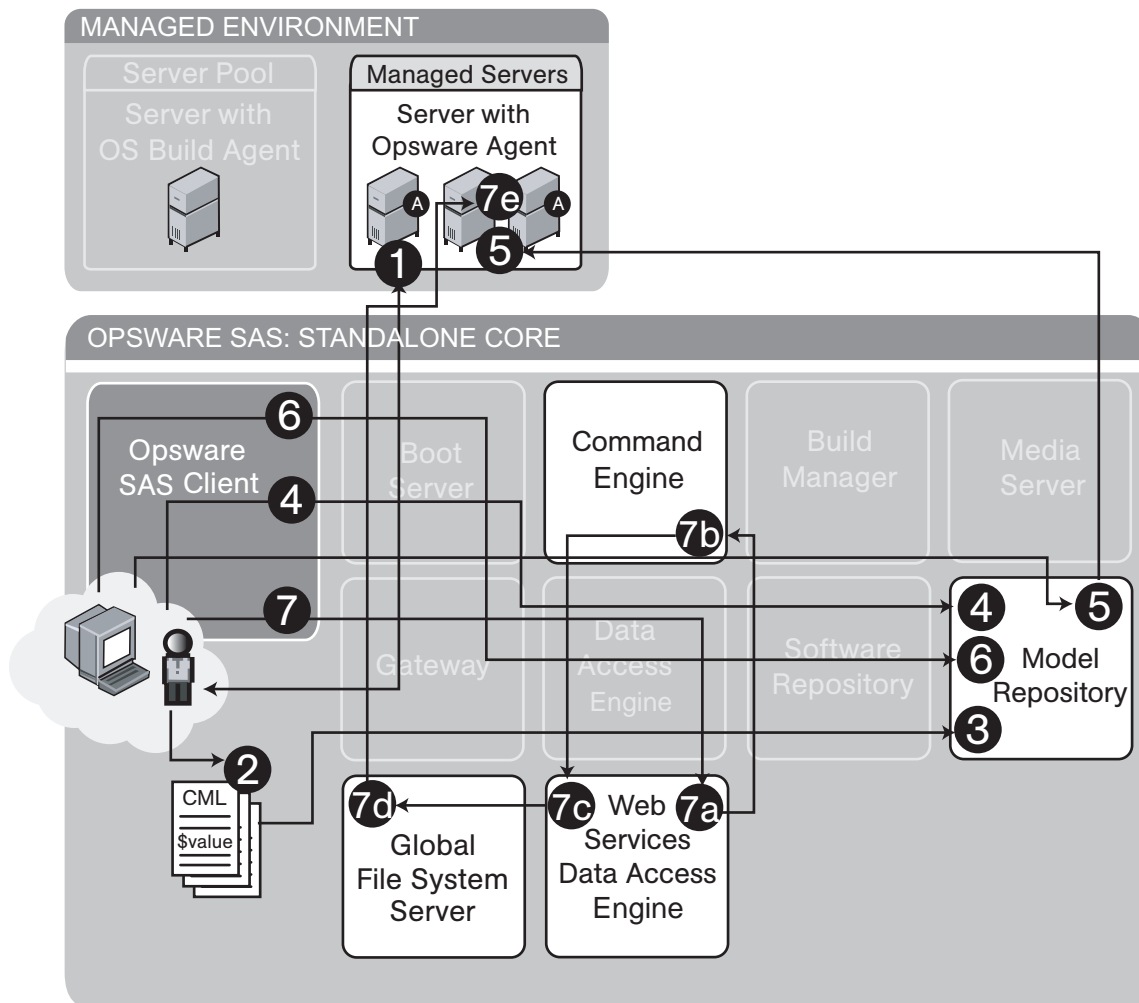


Figure 1-18 shows the process of discovering unmanaged servers and installing the Opware Agent on those servers:

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 SAS Client and creates an Application Configuration, which is stored in the Model Repository.

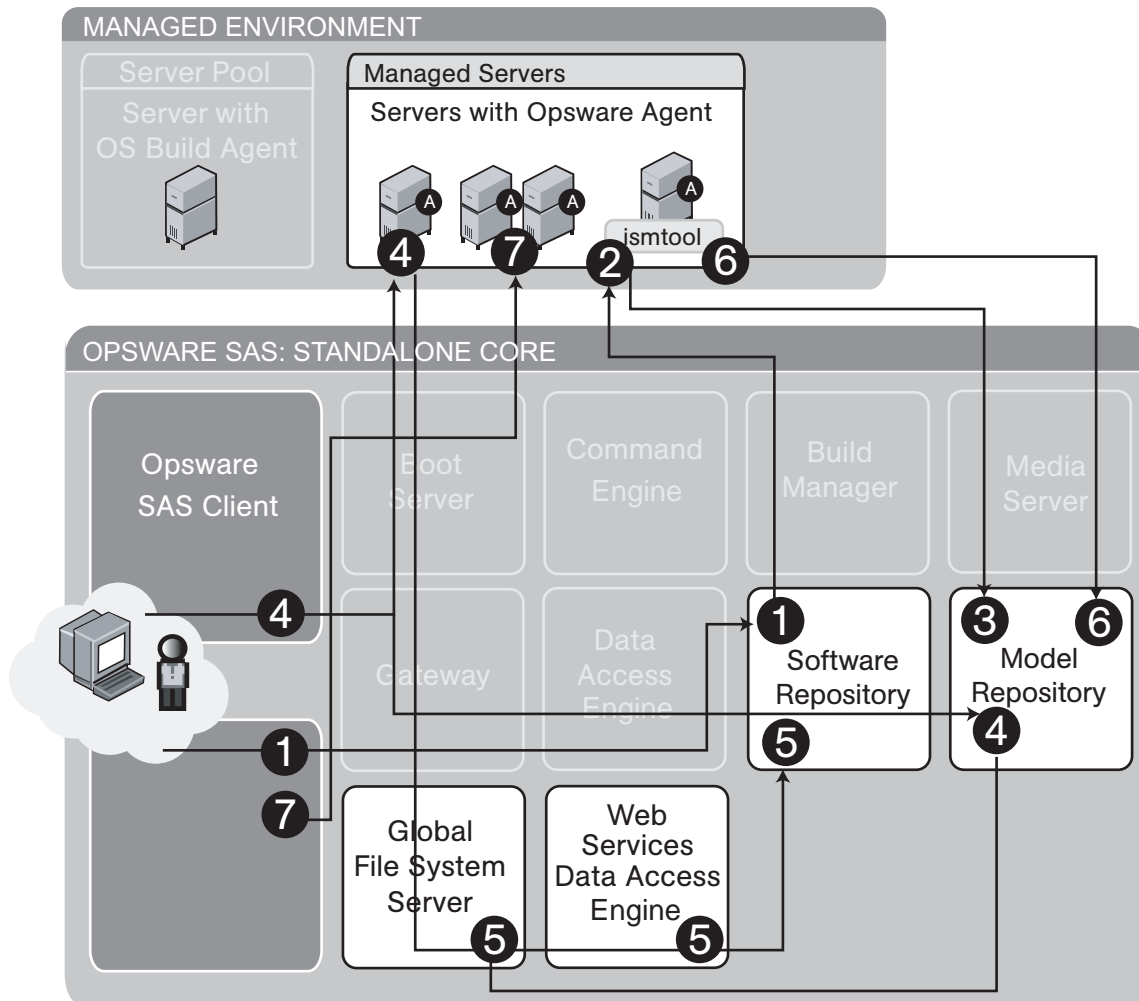
Part B: Configure and Push Application Configurations to Servers

- 5** The user chooses servers or server groups in the SAS 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 audit and remediation results 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 SAS Client and uses Managed Servers to select the server to use as a packaging server.

- 2** The user downloads and installs the ISM tool to prepare the Visual Packing server.
This will make the manage server a Visual Packaging server .
- 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 SAS 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

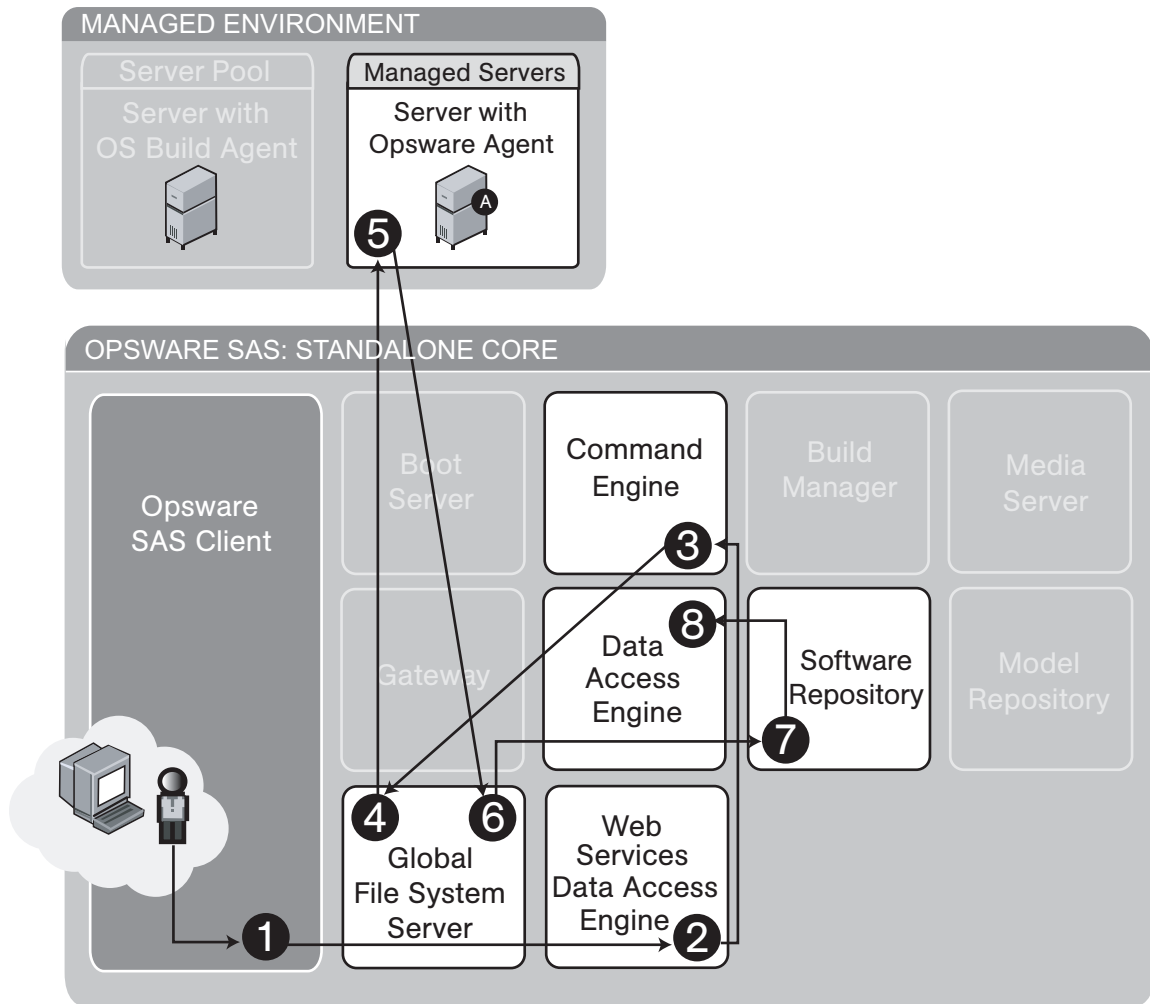
- 1** The user logs into the SAS Client and selects one of the following items as a source for the visual package:
 - A Server (from managed servers)
 - An Audit Result (from the Model Respository)
 - A Snapshot Result (from the Model Respository)
- 2** The user selects a location in the library for the Visual Packager. The user creates a software policy. 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.
- 3** The ismtool creates a software policy in the Library and attaches the new package.
- 4** In the SAS Client, the user attached the software policy containing the package and remediates the manage server to install the new visual package onto managed servers.

Audit and Remediation

The Opsware Audit and Remediation feature enables Opsware users to keep managed servers up-to-date by comparing them to known working servers.

Figure 1-20: Component Interaction of Taking Snapshots

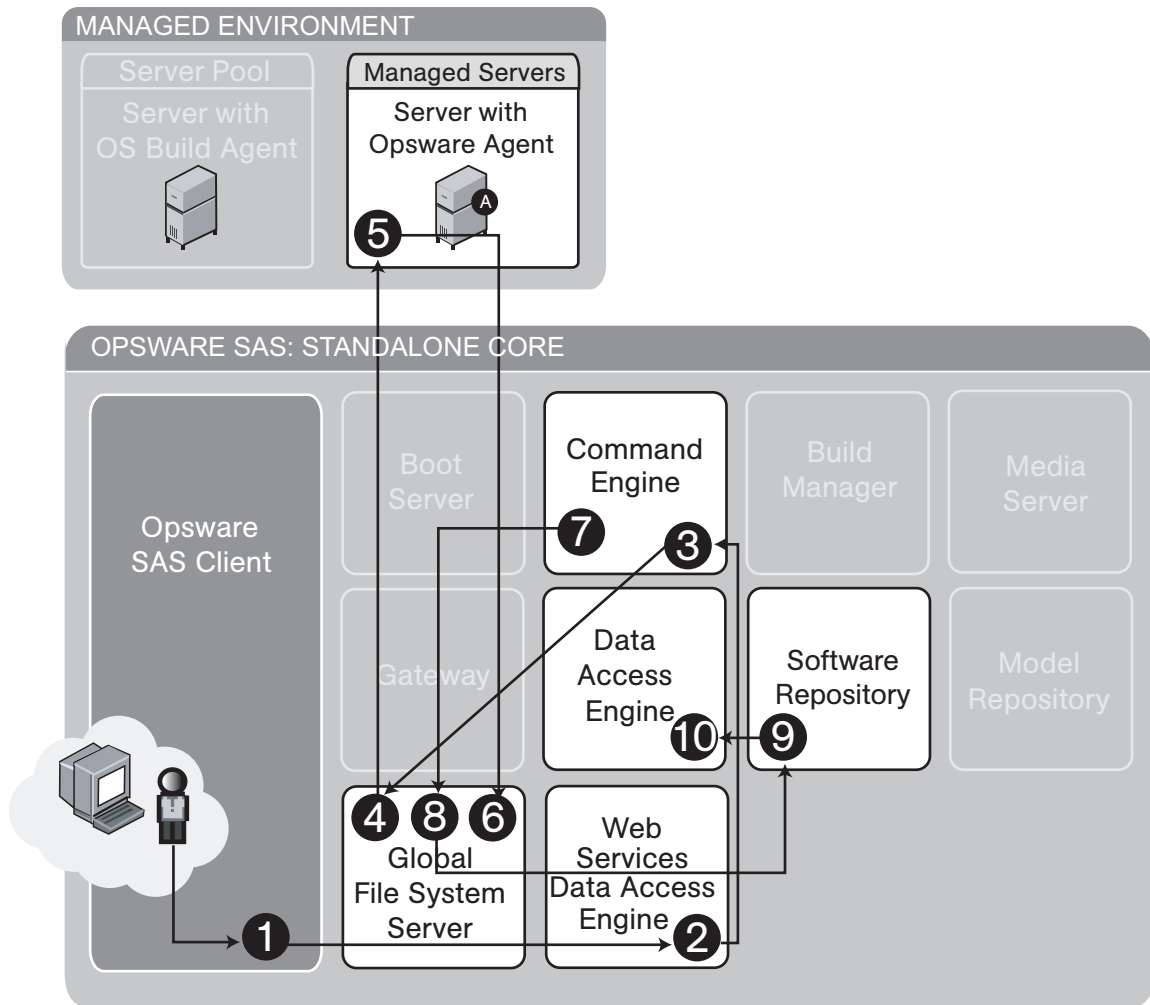


Audit and Remediation: Take a Snapshot

- 1 An Opsware user chooses a snapshot specification to run.
- 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.

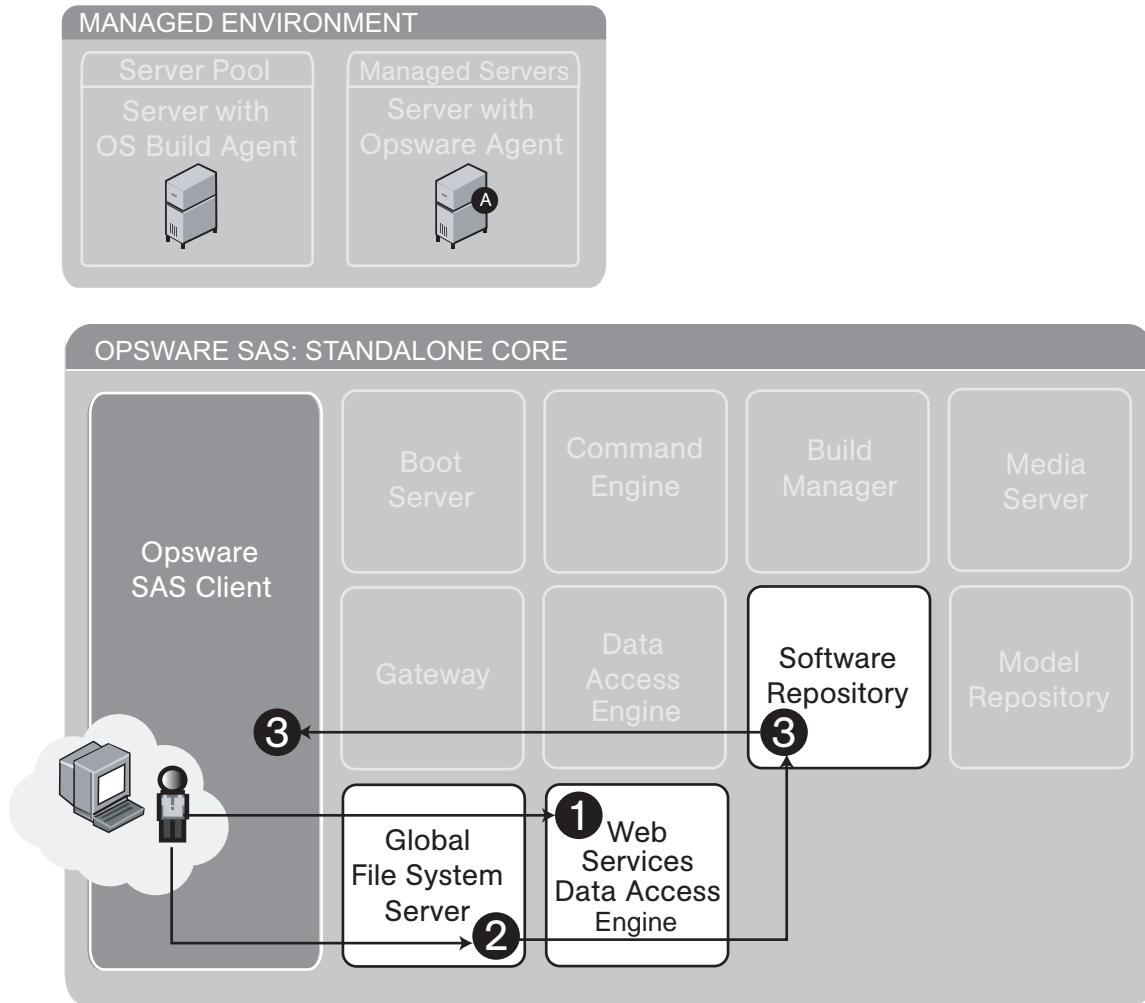
Figure 1-21: Component Interaction of Auditing a Server



Audit and Remediation: Run an Audit

- 1** An Opsware user chooses an audit 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



Audit and Remediation: 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: User and Group Setup

IN THIS CHAPTER

This section discusses the following topics:

- Users, Groups, and Permissions
- Managing Users
- Managing User Groups and Permissions
- Managing the Special Administrators Group
- External LDAP Directory Service with Opware SAS
- Code Deployment Permissions

Users, Groups, and Permissions

Opware SAS enforces a security policy that allows only authorized users to perform specific operations on specific servers. This section explains how to set up a role-based security structure for Opware SAS.

Opware Users and User Groups

When you log on to the SAS Web Client, you are prompted for an Opware user name and password. Everyone in your organization who logs into the SAS Web Client must have a unique Opware user name and password. User names are stored in the Model Repository. You can create user names with the SAS Web Client, or you can import them into the Model Repository from an external Lightweight Directory Access Protocol (LDAP) system. Opware user names are not case sensitive.

A user group represents a role played by the people in your organization who log on to the SAS Web Client. Every user should belong to one or more Opware user groups. The tasks that a user is authorized to perform depend on the groups the user belongs to. You define permissions for a user group, not for individual users.

Opware Permissions

The permissions that you specify for a user group determine what the group's members can do with Opware SAS. Feature permissions specify what actions users can perform; resource permissions indicate which objects (typically servers) users can perform these actions on. For example, Jane Doe could belong to a user group called London Windows Administrators. This user group has the feature permission to install patches, and the resource permission to Read & Write on the device group named London Windows Servers.

Feature Permissions

An Opware SAS feature is a task, such as running a script or uploading a patch. With feature permissions, you define the tasks that can be performed by the users of a group. A feature permission is either on or off: The user can either perform a task or cannot. In the SAS Web Client, you specify feature permissions on the Features, Client Features, and Others tabs of the Edit Group page.

Resource Permissions

A resource is usually a set of managed servers. A resource permission determines if the users in a user group can view or modify a resource. Resource permissions specify the following types of access:

- **Read:** Users can view the resource only.
- **Read & Write:** Users can view, create, modify or delete the resource.
- **None:** The resource does not appear in the Opware SAS Client or the SAS Web Client. Users cannot view or modify the resource.

The SAS Web Client organizes resources into the following categories:

- **Customers:** The servers associated with a customer.
- **Facilities:** The servers associated with a facility.
- **Device Groups:** The servers belonging to the specified public device group.

Each of the preceding resource categories corresponds to a tab on the Edit Group page of the SAS Web Client.

Managed servers are the most common resources. Other types of resources are application configurations, hardware definitions, realms, and OS installation profiles. Each of these resources can be associated with customers.

Folders can also be associated with customers, but the access to folders is controlled in a different way. (See “Folder Permissions” on page 74.)

Server Access and Resource Permissions

Access to a server depends on the server’s association to a customer, association to a facility, and optionally, its membership in a public device group. For example, suppose that a server is associated with the Widget Inc. customer, resides in the Fresno facility, and belongs to the Accounting device group. To modify the server, the user group must have the permissions listed in Table 2-1. (The Read & Write permission for Accounting is required only if user group permissions are specified for public device groups.)

Table 2-1: Example of Resource Permissions

RESOURCE	GROUP PERMISSION
Customer: Widget, Inc.	Read & Write
Facility: Fresno	Read & Write
Device Group: Accounting	Read & Write

If the permissions for the customer, facility, or device group do not match, then the most restrictive permissions are enforced. For example, if the permission for the Customer is Read & Write, but the permission for the facility is Read, then the Read permission is enforced. If the permission for the Customer is None, then the server cannot be viewed, even if the other permissions for the user group specify Read (or Read & Write).

Feature and Resource Permissions Combined

To use a feature on a resource, the user must belong to a group that has the necessary permissions for both the feature and resource. For example, suppose that a server is associated with these resources: the Widget, Inc. customer and the Fresno facility. To install a patch on this server, the user must belong to a group with the permissions listed in Table 2-2.

Table 2-2: Example of Permissions Resources and Features

RESOURCE OR FEATURE	GROUP PERMISSION
Customer: Widget, Inc.	Read & Write
Facility: Fresno	Read & Write
Feature: Install Patch	Yes

Folder Permissions

Folder permissions control access to the contents of the folder, such as software policies, packages, OS sequences, and subfolders. A folder's permissions apply only to the items directly under the folder. They do not apply to items lower down in the hierarchy, such as the subfolders of subfolders (grandchildren).

Types of Folder Permissions

In the Folders Properties window of the Opsware SAS Client, you can assign the following permissions to a user group:

- **List Contents of Folder:** Navigate to the folder in the hierarchy, click on the folder, view the folder's properties, see the name and type of the folder's children (but not the attributes of the children).
- **Read Objects Within Folder:** View all attributes of the folder's children, open object browsers on folder's children, use folder's children in actions.

For example, if the folder contains a software policy, users can open (view) the policy and use the policy to remediate a server. However, users cannot modify the policy. (For remediation, feature and server permissions are required, as well.)

Selecting this permission automatically adds the List Contents of Folder permission.

- **Write Objects Within Folder:** View, use, and modify the folder's children.

This permission permits actions such as New Folder and New Software Policy. To perform most actions, client features are required as well.

Selecting this permission automatically adds the List Contents of Folder and the Read Objects Within Folder permissions.

- **Edit Folder Permissions:** Modify the permissions or add customers to the folder.

This permission enables users to delegate the permissions management of a folder (and its children) to another user group.

Selecting this permission automatically adds the List Contents of Folder permission. To select this permission, the user must belong to the Administrators group.

Client Feature Permissions and Folders

Client feature permissions determine what actions users can perform with the Opsware SAS Client. Folder permissions specify which folders users have access to.

To perform most actions on folders and the items they contain, users need both folder and client feature permissions. For example, to add a software policy to a folder, users must belong to a group that has the Write Objects Within Folder permission and the Manage Software Policy permission (Read & Write).

Customer Constraints, Folders, and Software Policies

If a customer is assigned to a folder, the customer constrains some of the actions on the software policies contained in the folder. These constraints are enforced through filtering: The objects that can be associated with the software policies must have a matching customer.

For example, suppose that you want to add the `quota.rpm` package to a software policy. The package and the software policy reside in different folders. The customer of the policy's parent folder is Widget and the customer of the package's parent folder is Acme. When you perform the Add Package action on the policy, the packages that you can choose will not include `quota.rpm`. The customer of the policy's parent folder (Widget) acts as a filter, restricting the objects that can be added to the policy. If you add the Widget customer to the parent folder of `quota.rpm`, then you can add `quota.rpm` to the policy.

The following list summarizes the customer constraints for software policy actions. These constraints are invoked only if the software policy's parent folder has one or more customers. Software policy actions not listed here, such as New Folder, do not have customer constraints.

- **Add Package:** The customers of the package's parent folder must be a subset of the customers of the software policy's parent folder.
- **Add Application Configuration:** The customers of the application configuration must be a subset of the customers of the software policy's parent folder.
- **Add Software Policy:** If software policy A is added to software policy B, then the customers of A's parent folder must be a subset of the customers of B's parent folder.
- **Attach Software Policy:** The customer of the server being attached must be one of the customers of the software policy's parent folder.
- **Install Software Policy Template:** The customer of the server must be one of the customers of the parent folder of each software policy contained in the template.

Default Folder Permissions

When Opware SAS is first installed, the default user groups are assigned permissions to the top-level folders. For details, see “Predefined User Group Permissions” on page 303. When you create a new folder, it has the same permissions and customer as its parent.

Membership in Multiple Groups

If a user belongs to more than one user group, the user's permissions are derived from the resource and feature permissions of the groups. The way the permissions are derived depends on whether or not the resources are folders.

If the resources are not folders, then the derived permissions are a cross-product of the resource and feature permissions of all groups that the user belongs to. With a cross product, all feature permissions apply to all resource permissions. For example, Jane Doe belongs to both of the Atlanta and Portland groups, which have the permissions listed in Table 2-3. Because the derived permissions are a cross-product, Jane can perform the System Diagnosis task on the managed servers associated with the Widget Inc. customer, even though neither the Atlanta nor Portland group has this capability.

Table 2-3: Example of Cross-Product Permissions

RESOURCE OR FEATURE	ATLANTA USER GROUP PERMISSION	PORTLAND USER GROUP PERMISSION
Resource: Customer Widget, Inc.	Read & Write	None
Resource: Customer Acme Corp.	None	Read & Write
Feature: System Diagnosis	No	Yes

If the resources are folders (or their contents), then the derived permissions for the user are cumulative, but do not cross user groups. For example, Joe Smith belongs to both the Sunnyvale and Dallas groups shown in Table 2-4. Joe can create packages under the Webster folder because the Sunnyvale group has Read & Write permissions for that folder

and for the Manage Package feature. However, Joe cannot create packages under the Kiley folder, because neither user group can do so. Joe can create OS Sequences under the Kiley folder, but not under the Webster folder.

Table 2-4: Example of Cumulative Permissions

RESOURCE OR FEATURE	SUNNYVALE USER GROUP PERMISSION	DALLAS USER GROUP PERMISSION
Resource: Folder Webster	Read & Write	None
Resource: Folder Kiley	None	Read & Write
Feature: Manage Packages	Read & Write	None
Feature: Manage OS Sequences	None	Read & Write

Restricted Views of the SAS Web Client

The SAS Web Client displays only those features and resources that the user's group has Read (or Read & Write) permissions.

For example, John Smith belongs to the Basic Users group, which has the permissions listed in Table 2-5. When John logs in, the SAS Web Client displays only the servers for Widget Inc., but not those of Acme Corp. In the navigation panel of the SAS Web Client, the Operating Systems link appears, but not the Scripts link.

Table 2-5: Example of Permissions and Restricted Views

RESOURCE OR FEATURE	BASIC GROUP PERMISSION
Customer: Widget, Inc.	Read & Write
Customer: Acme Corp.	None
Wizard: Prepare OS	Yes
Wizard: Run Scripts	No

To locate or view a server, a user must belong to a group that has Read (or Read & Write) permission to both the customer and facility associated with the server. If the server also belongs to a device group with set permissions, then the user group must also have Read (or Read & Write) access to the device group. Otherwise, the user cannot locate the server in the SAS Web Client.

Predefined User Groups

Opware SAS includes the following predefined user groups:

- Basic Users
- Intermediate Users
- Advanced Users
- Opware System Administrators

The Basic, Intermediate, and Advanced Users groups define roles for system administrators with increasing levels of responsibility. These system administrators perform operational tasks on managed servers and set up elements of Opware SAS such as patches and packages. The users in the Opware System Administrators group manage Opware SAS itself, performing tasks such as running the Opware system diagnosis and multimaster tools.

Use of the predefined user groups is optional. You can change the permissions of the predefined user groups; you can also delete these groups. Changes or deletions of the predefined user groups are not affected by Opware SAS upgrades.

See “Predefined User Group Permissions” on page 303 in Appendix for more information.

Special Admin User and Administrators Group

The Opware Installer creates a single user called `admin` in the Administrators group for the initial login to the SAS Web Client. The password for `admin` is specified during the installation and should be changed immediately afterwards.



As a best practice, you should not add the `admin` user to other groups.

The users in the Administrators group manage the security structure of Opsware SAS. Members of the Administrators group create users and groups, specify permissions for groups, and assign users to groups. The Administrators group also has permissions to manage customers and facilities. Unlike the other groups in the SAS Web Client, the Administrators group has its own tab and cannot be deleted. (See Figure 2-4.) Also, you cannot change the permissions of the Administrators group.

The Administrators group manages the security of Opsware SAS, whereas the Opsware System Administrators group manages the components and infrastructure of Opsware SAS.

Process Overview for Security Administration

The person responsible for the security of Opsware SAS creates and maintains users, groups, and permissions. This person must be able to log on to the SAS Web Client as a user that belongs to the special Administrators group.

The following steps provide an overview of security administration for Opsware SAS:

- 1** Associate customers with managed servers. You may choose to use the initial customer that was specified during the installation of the Opsware core, or you may create new customers.
- 2** Create and name user groups to represent the roles of the people who will use the SAS Web Client. Alternatively, you may use the predefined user groups.
- 3** Set the resource permissions for the user groups. (These permissions are on the Customers, Facilities, and Device Groups tabs of the Edit Group page of the SAS Web Client.) If using the predefined user groups, you may need to add permissions for the customers that you created previously.
- 4** Delegate the setting of folder permissions to the user groups that will manage the folder hierarchy of the Opsware SAS Client.
- 5** Set the feature permissions for the user groups. (These permissions are on the Features, Client Features, and Other tabs of the SAS Web Client.)
- 6** Create new users or import existing users from an external LDAP.
- 7** Assign users to the appropriate groups.

Managing Users

To manage users, you must log on to the SAS Web Client as a user that belongs to the Administrators group. The default member of the Administrators group is the `admin` user.

Creating a User

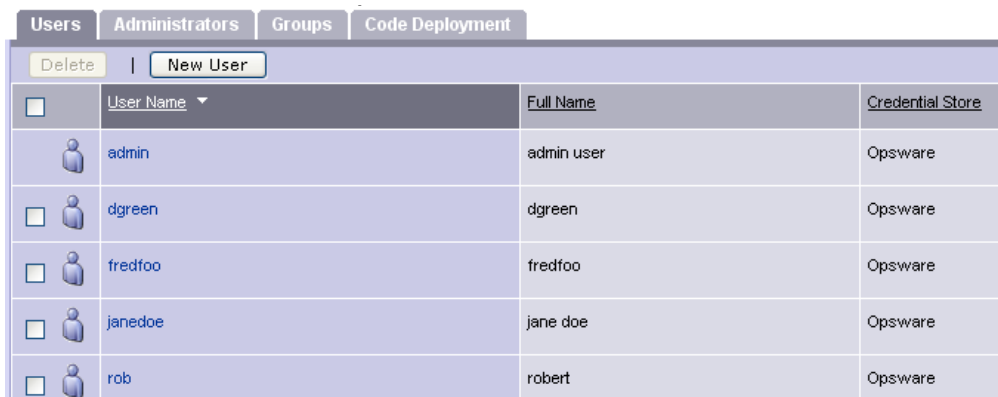
You can create Opware users with the SAS Web Client, or you can import users from an external LDAP directory. See “External LDAP Directory Service with Opware SAS” on page 93 in this chapter for more information.

To create a user with the SAS Web Client, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.

The Users tab appears. (See Figure 2-1.)

Figure 2-1: Users Tab



<input type="checkbox"/>	User Name	Full Name	Credential Store
<input type="checkbox"/>	admin	admin user	Opware
<input type="checkbox"/>	dgreen	dgreen	Opware
<input type="checkbox"/>	fredfoo	fredfoo	Opware
<input type="checkbox"/>	janedoe	jane doe	Opware
<input type="checkbox"/>	rob	robert	Opware

- 2 Click **New User**.
- 3 On the Profile Editor page, fill in the required fields; they're labeled in bold font.

The Login User Name may be different than the first, last, and full names. The Login User Name is not case sensitive and cannot be changed after the user is created.

Optionally, you may assign the user to one or more of the groups listed at the bottom of the page. Or, you may change the user's group membership at a later time. If a user does not belong to a group, the user cannot view servers or perform tasks with the SAS Web Client.

- 4 Click **Save** to create the user.

Editing User Information

Each Opsware user can edit the profile information for his or her own login user. To view or change information for your own login user, perform the following steps:

- 1 Click the My Profile link in the upper right corner of the SAS Web Client.
- 2 On the My Profile page, you can change most of the information displayed on the User Identification tab. You cannot change the user name (login user) or permissions. If the user name has been imported from an external LDAP directory, then the password cannot be changed with the SAS Web Client.

If your user belongs to the Administrators group, you may view or edit the information of any Opsware user. To do so, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 On the Users tab, select an entry in the User Name column.
- 3 In the Profile Editor, modify the information as appropriate.
- 4 Click **Save**.

Viewing a User's Permissions

You do not assign permissions directly to a user. Instead, you set the permissions on a user group and then assign a user to a group. To view the permissions of a user, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 On the Users tab, select an entry in the User Name column.
- 3 If the user belongs to more than one group, on the Edit User page, select a user group in the "View as" field. The permissions displayed depend on the user group you select.
- 4 View the permissions on the Resource Privileges and Action Privileges tabs.

Deleting a User

When you delete a user, the user's login and logout history is permanently stored, and the user is unassigned from user groups. After a user is deleted, you can create another user with the same name.

To delete an Opsware user, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.

- 2 On the Users tab, select the check box next to the user to be deleted.
- 3 Click **Delete**.

Managing User Groups and Permissions

To perform the tasks in this section, you must log on to the SAS Web Client as a user (such as `admin`) that belongs to the Administrators group. If you change permissions while a user is logged in to the SAS Web Client or SAS Client, the user must log out and log in again for the changes to take effect.

Creating a User Group

To create an Opware user group, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 On the Groups tab, click **New Group**.
- 3 On the New Group page, enter a role in the Group name field.
- 4 At this point, you can select the check boxes under the Feature column to assign permissions to the group. The New Group page does not display all available permissions.
- 5 Click **Save**.

Assigning a User to a Group

You should assign each Opware user to a group reflecting the user's role in your organization. If a user belongs to multiple user groups, the user has the permissions from all assigned groups. To assign an Opware user to a user group, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 On the Group tab, select a group from the Name column.
- 3 On the Users tab, in the Unassigned Members box, select the user name.
- 4 Click the right arrow.
- 5 To unassign a user, click the name in the Assigned Members box and click the left arrow.
- 6 Click **Save**.

Setting the Customer Permissions

In Opware SAS, you can associate a customer with a number of resources, including servers, folders, application configurations, and OS installation profiles. By setting the customer permission, you control the access that the users of a group have to the resources associated with the customer. For example, if you want the users of a group to be able to view (but not modify) the servers associated with the Widget Inc. customer, set the permission to Read.

The customer permissions also control access to the customer object itself. For example, to add a custom attribute to a customer, a user must belong to a group that has Read & Write permission to the specific customer, as well as permission for the Customers feature.

To control the access to the resources associated with a customer, perform the following steps:

- 1** From the navigation panel, select Administration ► Users & Groups.
- 2** On the Groups tab, select an entry in the Name column. Another set of tabs appears, including the Customers tab.
- 3** On the Customers tab, for each customer listed, select Read, Read & Write, or None.
- 4** Click **Save**.

Setting the Facility Permissions

In Opware SAS, a facility can be associated with resources such as servers and IP ranges. To modify a server of a particular facility, a user must belong to a group that has Read & Write permission for the facility.

The facility permissions also control access to the facility object itself. For example, to modify a property of a facility, a user must belong to a group that has Read & Write permission to the facility, as well as permission for the Facilities feature.

To control the access to the resources associated with a facility, perform the following steps:

- 1** From the navigation panel, select Administration ► Users & Groups.
- 2** On the Groups tab, select an entry in the Name column. Another set of tabs appears, including the Facilities tab.
- 3** On the Facilities tab, select Read, Read & Write, or None.

- 4 Click **Save**.

Setting the Device Group Permissions

To control access to the servers in a public device group, select a permission on the Device Groups tab. (You cannot control access to a private device group, which is visible only to the user who created it.)

If the Device Groups tab lists no device groups, then access to servers is not controlled by membership in device groups; however, access to servers is still controlled by their association with customers and facilities. If the Device Groups tab lists at least one device group, then access is denied to unlisted device groups (the equivalent of a None permission).

Access control based on device groups is optional. By default, membership in a device group does not restrict access. In contrast, for servers associated with customers or facilities, the default permission is None, which prohibits access.

You can combine customer, facility, and device group permissions to implement security policies. For example, you can restrict access to servers that are associated with the Acme Corp. customer, reside in the Fresno facility, and belong to a device group that contains only Windows servers.

A device group can contain other device groups. However, permissions are not inherited by the contained (children) device groups.

The permissions on the Device Groups tab control access to servers that belong to device groups. However, these permissions do not control the management of the device groups. To create, modify, or delete device groups, a user must belong to a user group that has the Manage Public Device Groups and the Model Public Device Groups check boxes selected on the Other tab. Also, the Managed Servers and Groups check box must be selected on the Features tab.

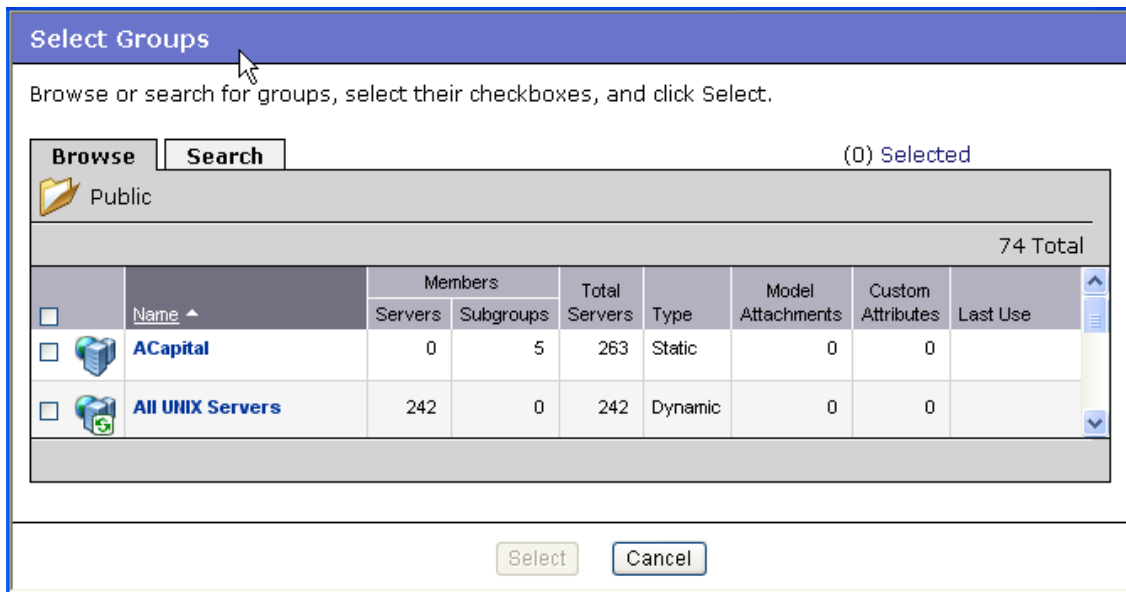
To control access to servers that belong to a device group, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 On the Groups tab, select an entry in the Name column. Another set of tabs appears, including the Device Groups tab.
- 3 On the Device Groups tab, note the check box below **Assign**. If this check box is selected, then access to managed servers is not based on device groups.
- 4 Deselect the check box below **Assign**.

- 5** Click **Assign**.

The Select Groups page appears. (See Figure 2-2.)

Figure 2-2: Select Groups Page



- 6** On the Select Groups page, use the Browse or Search tab to locate the device groups.
- 7** On the Browser or Search tab, click on the device group name and then click **Select**.
- 8** On the Device Groups tab, for each device group listed, select the check box and click the button for the appropriate access.

To allow viewing (but not modification) of the servers in a device group, select the Read permission. To allow both viewing and modification, select the Read & Write permission.

- 9** Click **Save**.

Setting the General Feature Permissions

The Features tab of the SAS Web Client includes many tasks, including managing the servers and running the wizards. If the check box for a feature is unselected, then the SAS Web Client does not display the related links in the navigation panel.

To allow the users in a group the ability to view and execute a task on the Features tab, perform the following:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 On the Group tab, select a group from the Name column.
- 3 Another set of tabs appears, including the Features tab. (See Figure 2-3.)

Figure 2-3: Features Tab

Users Customers Facilities Node Stacks Server Groups Features Client Features Other		
Save Cancel		
<input type="checkbox"/>	Feature	Description
<input checked="" type="checkbox"/>	Configuration Tracking	Tracking, backup and restore of configuration files
<input type="checkbox"/>	Configure Opware	Allow users to manage configurations for opware-specific products
<input checked="" type="checkbox"/>	Customers	Manage customers
<input checked="" type="checkbox"/>	DNS	Add, edit and delete domain name service entries
<input type="checkbox"/>	Data Center Intelligence Reports	Data Center Intelligence Reports
<input checked="" type="checkbox"/>	Facilities	Manage facilities
<input checked="" type="checkbox"/>	IP Ranges & IP Range Groups	Add and edit IP mappings
<input type="checkbox"/>	

- 4 On the Features tab, select the check box for each feature that should be enabled for the user group. To prevent (and hide) a feature, deselect the check box.
- 5 Click **Save**.

Setting the Opware SAS Client Features Permissions

The Client Features tab of the SAS Web Client lists permissions for the actions performed with the SAS Client. These actions are for features such as Application Configuration and Software Policy Management.

To set these permissions for the SAS Client, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 On the Group tab, select a group from the Name column. Another set of tabs appears, including the Client Features tab.
- 3 On the Client Features tab, select the appropriate permission buttons.
- 4 Click **Save**.

Setting the Other Features Permissions

The Other tab of the SAS Web Client contains the following permissions:

- **General Permissions:** Allows users in a user group to edit shared scripts or run “my scripts” as root. The Features tab also has script-related permissions: Scripts, and Wizard: Run Scripts.
- **Server and Device Group Permissions:** Enables users in a user group to perform particular tasks on managed servers. The Allow Run Refresh Jobs permission lets users specify a job to update the servers list. The Manage Public Servers Group permission enables users to create device groups, modify the group properties, and change the group membership (through rule changes, or adding and deleting servers). All users may view all public device groups. The Model Public Servers Group permission lets users add custom attributes. (These permissions apply to public, not private device groups. Only the user who creates a private device group can view or modify it.) The Features tab also has a permission related to managing servers: Managed Servers and Group.
- **Job Permissions:** Allows users in a user group to view and schedule jobs, which include operations such as Audit Servers, Snapshots, Push Configurations, and Audit Configurations. The View All Jobs permission lets users view the details and schedules of jobs created by all users. The Edit All Jobs permission enables users to view or modify the schedules of jobs created by all users and to view the job details of all users. Without these permissions, users can view and schedule only their own jobs.

To set the permissions on the Other tab, perform the following steps:

- 1** From the navigation panel, select Administration ► Users & Groups.
- 2** On the Group tab, select a group from the Name column. Another set of tabs appears, including the Other tab.
- 3** On the Other tab, select the check boxes to assign permissions to this user group.
- 4** Click **Save**.

Setting the Permissions for the Opsware Global Shell File System

The Opsware Global Shell File System (OGFS) underlies many SAS Client features, such as browsing managed server file systems and scanning servers for compliance. For features that access the OGFS, the necessary permissions are granted with the `aaa` command of the Global Shell.

For example, to create a Snapshot Specification with rules that include the files of a managed server, the `readServerFilesystem` permission must be granted with the `aaa` command.

To set up permissions for the OGFS, perform the following steps:

- 1** Log on to the Opware SAS Web Client as a user that belongs to the Administrators group.
- 2** Launch the Opware SAS Client.
- 3** Open a Global Shell session.
- 4** Run the `aaa` command-line utility to specify the operations that can be performed within a Global Shell session by the members of a user group.

The following example gives all members of the Alpha group permission to open a Global Shell session:

```
aaa shell-perm grant -o launchGlobalShell -u Alpha
```

For syntax and more examples of the `aaa` utility, see the *Opware® SAS User's Guide: Server Automation*.

Setting Folder Permissions

When a folder is created, it has the same permissions (and customer) as its parent folder. If you are changing the permissions of a folder that has children, you are prompted to apply the changes to the children. To set the permissions of a folder, perform the following steps:

- 1** In the Opware SAS Client, navigate to the folder.
- 2** From the Actions menu, select **Folder Properties**.
- 3** In the Folder Properties window, select the Permissions tab.
- 4** On the Permissions tab, click **Add** to allow certain user groups to access the folder.
- 5** For each user group displayed on the Permissions tab, select a check box such as List Contents of Folder or Write Objects Within Folder.

Delegating Folder Permissions

Every folder has a set of permissions. In large organizations, the security administrator might not have time to set the permissions on many folders. In this case, the security administrator can delegate the setting of folder permissions to the policy setters who are responsible for different applications. Only members of the Administrators group can delegate folder permissions.

To delegate folder permissions, perform the following steps:

- 1** In the SAS Web Client, assign the client feature Manage Folders to the user group of the policy setters.
- 2** In the Opware SAS Client, follow the instructions in “Setting Folder Permissions” on page 88.
- 3** On the Permissions tab, select (or add) the user group that the policy setter belongs to.
- 4** Select Edit Folder Permissions.

This selection delegates the setting of permissions for this folder, enabling the policy setters to control access to this folder and its children.

- 5** If you want to allow the policy setter to create subfolders, then select Write Objects Within Folder.

This selection enables the policy setter to create folder hierarchies beneath the current folder.

Managing the Special Administrators Group

To manage the Administrators group, users must belong to the Administrators group. Only those users who belong to the Administrators group can manage Opware SAS users and user groups. You can change user membership of the Administrators group, but you cannot modify the group in any other way. See “Special Admin User and Administrators Group” on page 78 in this chapter for more information.

Adding a User to the Administrators Group

To add a user to the Administrators group, perform the following steps:

- 1** From the navigation panel, select Administration ► Users & Groups.

- 2 Select the Administrators tab. (See Figure 2-4.)

Figure 2-4: Administrators Tab

<input type="checkbox"/>	User Name	Full Name
<input type="checkbox"/>	admin	admin user
<input type="checkbox"/>	dgreen	dgreen
<input type="checkbox"/>	rob	robert

- 3 Click **New Administrator**.
- 4 On the Add Administrators page, select the user you want to add to the Administrators group.
- 5 Click **Save**.

Removing a User from the Administrators Group

To remove a user from the Administrators group, perform the following steps:

- 1 From the navigation panel, select Administration ► Users & Groups.
- 2 Select the Administrators tab.
- 3 Select the check box for the user.
- 4 Click **Revoke**.

Password Policy Parameters

The Opware administrator can enable and configure the password policy parameters for accessing the SAS Web Client. The passwords will be checked against the configured parameters when user accounts are created by the Opware administrator or when the passwords are changed by the users or the administrator. The users, including the administrators will be alerted with an error message if their password does not match the criteria specified in the configured password policy parameters.

The Opware administrator can use the Administration features of the SAS Web Client to enable and configure the following parameters in a password:

- Set the maximum number of consecutive repeating characters allowed for a password. By default the value is 2, and the value cannot be 0.
- Set the minimum character limit required for a password. By default the minimum character limit is 6 and the maximum character limit is 50.
- Set the minimum non-alphabetic character limit required for a password. By default the minimum non-alphabetic character limit is 0 and the value cannot be greater than the value specified for the minimum character limit required for a password.

The Opware administrator can configure any number of the three password policy parameters for accessing the SAS Web Client. If the password policy parameter is disabled the password will be checked to ensure that it has at least 6 characters.

Enabling and Configuring Password Policy Parameters

Perform the following steps to enable and configure the password policy parameters for accessing the SAS Web Client.

- 1** Log on to the SAS Web Client as a user with admin privileges with the password you supplied during the interview. The SAS Web Client home page appears.
- 2** From the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 3** Under Select a Product, click the SAS Web Client link. The Modify Configuration Parameters for the SAS Web Client page appears.
- 4** To enable the password policy parameters, in the field, `owm.features.Min>PasswordPolicy.allow`, enter `true` as Figure 2-5 shows. The default value is `false`.

Figure 2-5: Enabling Password Policy Parameters

owm.features.Min>PasswordPolicy.allow:
Allow Password Policy features in OCC. (valid value: true, false)

Use default value: *no value*
 Use value:

- 5 In the field, `owm.pwpolicy.maxRepeats`, enter a value specifying the maximum number of consecutive repeating characters allowed for a password. The value entered must be greater than 0; the default value is 2. See Figure 2-6.

Figure 2-6: Configuring Maximum Number of Repeating Characters for a Password

<p>owm.pwpolicy.maxRepeats: Maximum number of same consecutive characters in password. (valid value: 1 or more)</p>	<p><input type="radio"/> Use default value: <i>no value</i></p> <p><input checked="" type="radio"/> Use value: <input type="text" value="1"/></p>
--	---

- 6 In the field, `owm.pwpolicy.minChars`, enter a value specifying the minimum number of characters required for a password. The value must be a positive integer; the default value is 6. See Figure 2-7.

Figure 2-7: Configuring Minimum Number of Characters for a Password

<p>owm.pwpolicy.minChars: Minimum number of characters for password. (valid value: positive integer. Note: if a value less than 6 is specified, a 6 character password is enforced.)</p>	<p><input type="radio"/> Use default value: <i>no value</i></p> <p><input checked="" type="radio"/> Use value: <input type="text" value="10"/></p>
---	--

- 7 In the field, `owm.pwpolicy.minNonAlphaChars`, enter a value specifying the minimum number of non-alphabetic characters required for a password. The value cannot be greater than the value specified for the minimum character limit; the default value is 0. See Figure 2-8.

Figure 2-8: Configuring Non-alphabetic Characters for a Password

<p>owm.pwpolicy.minNonAlphaChars: Minimum number of non-alphabetic characters in password. (valid value: 0 or more)</p>	<p><input type="radio"/> Use default value: <i>no value</i></p> <p><input checked="" type="radio"/> Use value: <input type="text" value="3"/></p>
--	---

- 8 Click **Save** to apply the changes. The configuration page refreshes and a message appears that the update was successful.



When you make changes to the password policy System Configuration settings in one core of a mesh, the change is reflected in that core only. The changes get propagated to other cores in a mesh only after you restart the other cores.

Disabling Password Policy Parameters

Perform the following steps to disable the password policy parameters for accessing the SAS Web Client.

- 1** Log on to the SAS Web Client as an user with admin privileges with the password you supplied during the interview. The SAS Web Client home page appears.
- 2** From the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 3** Under Select a Product, click the SAS Web Client link. The Modify Configuration Parameters for the SAS Web Client page appears.
- 4** To disable the password policy parameters, in the field, `owm.features.MiniPasswordPolicy.allow`, select the default value (`false`) as Figure 2-9 shows. If you select the default value, the password will be checked to ensure that it has at least 6 characters.

Figure 2-9: Disabling Password Policy Parameters

owm.features.MiniPasswordPolicy.allow:
Allow Password Policy features in OCC. (valid value: true, false)

Use default value: `no value`

Use value:

- 5** Click **Save** to apply the changes. The configuration page refreshes and a message appears that the update was successful.



When you make changes to the password policy System Configuration settings in one core of a mesh, the change is reflected in that core only. The changes get propagated to other cores in a mesh only after you restart the other cores.

External LDAP Directory Service with Opware SAS

You can configure Opware SAS to use an external LDAP directory service for user authentication. With external authentication, you do not have to maintain separate user names and passwords for Opware SAS. When users log on to the SAS Web Client, they enter their LDAP user names and passwords.

Imported Users

With the SAS Web Client, you search for users in the external LDAP and then you import selected users into Opware SAS. You can limit the search results by specifying a filter.

The import process fetches the following user attributes from the LDAP:

```
firstName
lastName
fullName
emailAddress
phoneNumber
street
city
state
country
```

After the import process, you may edit the preceding list of attributes with the SAS Web Client. However, you cannot change the user login name or password with the SAS Web Client. Importing a user is a one-time, one-way process. Changes to the user attributes you make using the SAS Web Client are not propagated back to the external LDAP directory server, and vice versa.

Imported users are managed in the same way as users created by the SAS Web Client. For example, you use the SAS Web Client to assign imported users to user groups and to delete imported users from Opware SAS. If you delete an imported user with the SAS Web Client, the user is not deleted from the external LDAP directory.

If you use external authentication, you can still create separate users with the SAS Web Client. However, this practice is not recommended. On the User tab of the SAS Web Client, the Credential Store column identifies the source (External or Opware) of each user.

SSL and External Authentication

Although SSL is not required for external authentication, it is strongly recommended. The certificate files needed for LDAP over SSL must be in Privacy Enhanced Mail (PEM) format. Depending on the LDAP server, you may need to convert the server's CA certificate to PEM format.

Supported External LDAP Directory Servers

The following directory server products may be used with Opware SAS:

- Microsoft Active Directory (Windows 2000 or Windows 2003)
- Novell eDirectory 8.7

- SunDS 5.2

Using an LDAP Directory Server with Opsware SAS

To use an LDAP directory server with Opsware SAS, perform the following basic steps:

- 1** Add the `aaa.ldap` entries to the `twistOverrides.conf` file with a text editor. See “Modifying the Web Services Data Access Engine Configuration File” on page 95.
- 2** Get the SSL server certificate from the LDAP directory server. See “Importing a Server Certificate from the LDAP into Opsware SAS” on page 99. (Use of SSL is not required, but strongly recommended.)
- 3** Edit the `loginModule.conf` file with a text editor. See “Configuring the JAAS Login Module (`loginModule.conf`)” on page 100.
- 4** Restart the Web Services Data Access Engine:

```
/etc/init.d/opsware-sas restart twist
```
- 5** Use the SAS Web Client to import users from the LDAP directory server into Opsware SAS. See “Importing External LDAP Users” on page 101.

In a multimaster mesh, you must perform steps 1 - 4 on each Web Services Data Access Engine.

Modifying the Web Services Data Access Engine Configuration File

To modify `twistOverrides.conf`, perform the following steps:

- 1** Log in as root to the system running the Web Services Data Access Engine, an Opsware core component.
- 2** In a text editor, open this file:

```
/etc/opt/opsware/twist/twistOverrides.conf
```
- 3** In the text editor, add the necessary properties (listed in Table 2-6) to the `twistOverrides.conf` file. Although not required, the SSL properties are recommended. For examples of the lines required for the `twistOverrides.conf` file see, the sections that follow Table 2-6.
- 4** Save the `twistOverrides.conf` file and exit the text editor.

- 5** Make sure that the Unix `twist` user has write access to the `twistOverrides.conf` file.

Table 2-6: Properties in `twistOverrides.conf` for an External LDAP

PROPERTY	DESCRIPTION
<code>aaa.ldap.hostname</code>	The host name of the system running the LDAP directory server.
<code>aaa.ldap.port</code>	The port number of the LDAP directory server.
<code>aaa.ldap.search.binddn</code>	The BIND DN (Distinguished Name) for LDAP is required by the search of the import user operation. A blank value denotes an anonymous BIND.
<code>aaa.ldap.search.pw</code>	The BIND password for LDAP is required by the search for the import user operation. A blank value denotes an anonymous BIND.
<code>aaa.ldap.search.filter.template</code>	The search filter template is used, with optional filter substitution, as the filter in the LDAP search for the user import. Any dollar sign (\$) character in the template will be replaced by the filter string specified in the Import Users page of the SAS Web Client. (The default value is an asterisk (*) which matches all entries.)
<code>aaa.ldap.search.base.template</code>	The configurable template allows support for a range of DIT configurations and schema in the LDAP service. The search base template string is used for the “search base” in the LDAP search operations for the user import.
<code>aaa.ldap.search.naming.attribute</code>	The naming attribute allows support for a range of schema in the LDAP services. Some use <code>uid</code> , others use <code>cn</code> , and so on. The value of this attribute is used for the internal user ID in Opsware SAS.

Table 2-6: Properties in `twistOverrides.conf` for an External LDAP (continued)

PROPERTY	DESCRIPTION
<code>aaa.ldap.search.naming.display.name</code>	The naming attribute allows support for a range of schema in the LDAP services. Some use <code>cn</code> , others use <code>displayName</code> , and so on. The value of this attribute is used for the Full Name of Opsware SAS user.
<code>aaa.ldap.ssl</code>	SSL: A value of <code>true</code> enables SSL.
<code>aaa.ldap.secureport</code>	SSL: The secure port of the LDAP directory server.
<code>aaa.ldap.usestarttls</code>	SSL: A value of <code>true</code> enables Start TLS.
<code>aaa.ldap.servercert.ca.fname</code>	SSL: The fully qualified file name of the server CA certificate.
<code>aaa.ldap.clientcert</code>	SSL: A value of <code>true</code> enables client certificate use.
<code>aaa.ldap.clientcert.fname</code>	SSL: The fully qualified file name of the client certificate.
<code>aaa.ldap.clientcert.ca.fname</code>	SSL: The fully qualified file name of the client CA certificate.

Example: `twistOverrides.conf` for Microsoft Active Directory Without SSL

```

aaa.ldap.search.binddn=cn=Administrator,cn=users,dc=example,dc=com
aaa.ldap.search.pw=secret
aaa.ldap.hostname=myservername.internal.example.com
aaa.ldap.port=389
aaa.ldap.search.filter.template=( &(objectclass=user)(cn=$) )
aaa.ldap.search.base.template=cn=users,dc=example,dc=com
aaa.ldap.search.naming.attribute=samaccountname
aaa.ldap.search.naming.display.name=cn

```

Example: `twistOverrides.conf` for Microsoft Active Directory With SSL

```

aaa.ldap.search.binddn=cn=Administrator,cn=users,dc=example,dc=com

```

```
aaa.ldap.search.pw=secret
aaa.ldap.hostname=myservername.internal.example.com
aaa.ldap.secureport=636
aaa.ldap.ssl=true
aaa.ldap.servercert.ca.fname=/var/opt/opsware/crypto/twist/
cert.pem
aaa.ldap.search.filter.template=( &(objectclass=user) (cn=$) )
aaa.ldap.search.base.template=cn=users,dc=example,dc=com
aaa.ldap.search.naming.attribute=samaccountname
aaa.ldap.search.naming.display.name=cn
```

Example: twistOverrides.conf for Novell eDirectory Without SSL

```
aaa.ldap.search.binddn=cn=admin,o=example
aaa.ldap.search.pw=secret
aaa.ldap.hostname=myservername.internal.example.com
aaa.ldap.port=389
aaa.ldap.search.filter.template=( &(objectclass=inetorgperson) (u
id=$) )
aaa.ldap.search.base.template=o=example
aaa.ldap.search.naming.attribute=uid
aaa.ldap.search.naming.display.name=cn
```

Example: twistOverrides.conf for Novell eDirectory With SSL

```
aaa.ldap.search.binddn=cn=admin,o=example
aaa.ldap.search.pw=secret
aaa.ldap.hostname=myservername.internal.example.com
aaa.ldap.secureport=636
aaa.ldap.ssl=true
aaa.ldap.servercert.ca.fname=/var/opt/opsware/crypto/twist/
ldapcert.pem
aaa.ldap.search.filter.template=( &(objectclass=inetorgperson) (u
id=$) )
aaa.ldap.search.base.template=o=example
aaa.ldap.search.naming.attribute=uid
aaa.ldap.search.naming.display.name=cn
```

Example: twistOverrides.conf for SunDS Without SSL

```
aaa.ldap.search.binddn=cn=Directory Manager
aaa.ldap.search.pw=secret
aaa.ldap.hostname=myservername.internal.example.com
aaa.ldap.port=389
aaa.ldap.search.filter.template=( &(objectclass=inetorgperson) (u
id=$) )
aaa.ldap.search.base.template=ou=people,dc=example,dc=com
aaa.ldap.search.naming.attribute=uid
```

```
aaa.ldap.search.naming.display.name=cn
```

Example: twistOverrides.conf for SunDS With SSL

```
aaa.ldap.search.binddn=cn=Directory Manager
aaa.ldap.search.pw=secret
aaa.ldap.hostname=myservername.internal.example.com
aaa.ldap.secureport=636
aaa.ldap.ssl=true
aaa.ldap.servercert.ca.fname=/var/opt/opsware/crypto/twist/
ldapcert.pem
aaa.ldap.search.filter.template=(&(objectclass=inetorgperson)(u
id=$))
aaa.ldap.search.base.template=ou=people,dc=example,dc=com
aaa.ldap.search.naming.attribute=uid
aaa.ldap.search.naming.display.name=cn
```

Importing a Server Certificate from the LDAP into Opware SAS

For SSL, the necessary certificates must be extracted from the LDAP and copied over to Opware SAS.

To import a server certificate from the LDAP into Opware SAS, perform the following steps:

- 1** Extract the server certificate from the external LDAP. For instructions, see the following sections.
- 2** Convert the extracted certificate to PEM format.

Certificates created on Windows systems are in Distinguished Encoding Rules (DER) format. The following example converts a certificate from DER to PEM format with the `openssl` utility:

```
OpenSSL> x509 -inform DER -outform PEM -in mycert.der \
-out mycert.pem
```

- 3** Copy the server certificate to the location specified by the Web Services Data Access Engine configuration file (`twistOverrides.conf`). For example, the `twistOverrides.conf` file could have the following line:

```
aaa.ldap.servercert.ca.fname=/var/opt/opsware/crypto/twist/
ldapcert.pem
```

Extracting the Server Certificate from Microsoft Active Directory

To extract the server certificate, perform the following steps:

- 1** Run either the Certificates MMC snap-in console or the Certificate Services web interface.
- 2** Export the Root CA cert from the Windows CA into DER format.

Extracting the Server Certificate from Novell eDirectory

To extract the server certificate, perform the following steps:

- 1** Find out the name of the local CA entry. (Example: CN=CORP-TREE CA.CN=Security)
- 2** Open the eDirectory Administration utility and click **Modify Object**.
- 3** Enter the entry name (CN=CORP-TREE CA.CN=Security).
- 4** Select the Certificates tab.
- 5** Click **Self Signed Certificate**.
- 6** Click **Export**.
- 7** In the dialog, click **No** for exporting the private key and then click **Next**.
- 8** Select the appropriate format (usually DER).
- 9** Click **Save the exported certificate to a file**.

Extracting the Server Certificate from SunDS

Typically, instead of exporting a server CA certificate from SunDS, you obtain the certificate that was imported into SunDS.

Configuring the JAAS Login Module (loginModule.conf)

To configure the JAAS login module, perform the following steps:

- 1** Log in as root to the system running the Web Services Data Access Engine, an Opsware core component.
- 2** In a text editor, open this file:
`/etc/opt/opsware/twist/loginModule.conf`
- 3** In the text editor, modify the `loginModule.conf` file so that it contains the following lines:

```
/** Login configuration for JAAS modules **/
```

```
TruthLoginModule {
  com.opsware.login.TruthLoginModule sufficient debug=true;
  com.opsware.login.LdapLoginModule sufficient debug=true;
};
```

- 4** Save the `loginModule.conf` file and exit the text editor.

Importing External LDAP Users

Before importing external LDAP users, you must complete the prerequisite steps. See “Using an LDAP Directory Server with Opsware SAS” on page 95 in this chapter for more information. After you import the users, the users may log on to the SAS Web Client with their LDAP user names and passwords.

To import external users, perform the following steps:

- 1** In the SAS Web Client, from the navigation panel, select Administration ► Users & Groups.
- 2** Select the Users tab. The page lists the existing Opsware SAS users.
- 3** On the Users tab, click **Import External Users**.

The page displays the users in the LDAP that match the search filter. The default filter is an asterisk (*), indicating that all users are selected. If a check box does not appear to the left of the user name, then the user already exists in Opsware SAS and cannot be imported.

If Opsware SAS cannot connect to the LDAP, check for error messages in the following file:

```
/var/log/opsware/twist/stdout.log
```

- 4** To change the search filter, enter a value in the field to the left of **Change Filter**. For example, to fetch only those user names beginning with the letter A, you enter A* in the field.
- 5** If you modified the search filter in the preceding step, click **Change Filter**. The page displays the users in the LDAP that match the search filter.
- 6** You can assign users to the user groups listed at the bottom of the page or you can assign them later.
- 7** Select the check boxes for the users you want to import. To import all users displayed, select the top check box.
- 8** On the Import Users page, click **Import**.

Code Deployment Permissions

Permissions to perform CDR operations are based on user membership in user groups predefined specifically for CDR. Users must also have the necessary permissions for the customer associated with the servers. Except for the Super User group, CDR operations are customer specific. A member of the Super User group can perform CDR operations on the servers of any customer.



The SAS Web Client might still show the legacy term CDS. However, all documentation references use Opsware SAS Code Deployment & Rollback term CDR.

The SAS Web Client includes predefined user groups that have specific permissions to perform CDR operations. Opsware administrators create and add users to these user groups to grant them permissions to perform specific CDR operations, based on their role in an organization. When logged into the SAS Web Client, users see only the services, synchronizations, and sequences that they have authorization to perform because of their user group membership. Users are assigned to these groups as part of the Create User process.

See “Code Deployment User Groups” on page 308 in Appendix for more information.

See the *Opsware® SAS User’s Guide: Server Automation* for information about the process to deploy code and content to managed servers.



When a user requests a service operation, synchronization, or sequence, an e-mail notification is sent to the individuals assigned to actually perform the requested service operation or synchronization.

Adding Members to a Code Deployment User Group

Permissions to perform specific Code Deployment operations are granted based on a user’s membership in specific Code Deployment user groups.

- 1** From the navigation panel, select Administration ► Users & Groups. The Manage Users: View Users page appears.
- 2** Select the Code Deployment tab.

- 3 Select the code deployment user group that you want to modify by clicking the hyperlinked user group name.

The Users and Groups: Edit Code Deployment Group - [group name] page appears.

- 4 From the drop-down list, choose the customer whose group membership you want to modify.



Code Deployment permission is assigned based on an Opsware customer. You cannot select Customer Independent, Not assigned, and Opsware customers and modify their group membership.

- 5 To add a user to the group, select the name in the left box, and then click the right arrow.

- 6 Click **Save** when you finish moving the user names to the box on the right.

A confirmation page appears.

- 7 Click **Continue**.

The Users & Groups: View Code Deployment Group page appears. You can continue modifying Code Deployment Groups, or you can select another function.

Chapter 3: Multimaster Mesh Administration

IN THIS CHAPTER

This section discusses the following topics:

- Overview of Opsware 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 Opsware Multimaster Mesh



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

A multimaster mesh is a set of Opsware 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 SAS Web Client is still usable from other facilities. Users in other facilities can have their own SAS Web Client. 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 SAS Web Client, 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 SAS Web Client 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 SAS Web Client, 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.

- 2 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 3-1 shows.

Figure 3-1: 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:	 Customer Independent  MYCUSTOMER	
<input type="button" value="Save"/> <input type="button" value="Cancel"/>		

- 3 To change the name of the facility that appears in the SAS Web Client, edit the Name field or click the Return to Facilities link to exit without making any changes.



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

- 4 Click **Save**. The SAS Web Client displays a message that confirms that the properties for that facility were updated.
- 5 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 SAS Web Client 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 200 in Chapter 5 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 SAS Web Clients
- The propensity for users to make changes in more than one facility by using different SAS Web Clients

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 SAS Web Client. Resolving the conflict unlocks the rows. See “Best Practices for Resolving Database Conflicts” on page 114 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 SAS Web Clients in different facilities. Conflicts occur when a user makes a change by using the SAS Web Client in one facility and another user makes a change to the same object using the SAS Web Client 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 Opsware 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. Opsware 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 SAS Web Clients, or between an SAS Web Client 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 Opsware Command Line Interface (OCLI) to upload the package `carol.conf`.
- 2 In the Phoenix facility, Carol logs into the SAS Web Client 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 SAS Web Client in Phoenix.

When the data arrives from Seattle, Opsware 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 Opsware 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 software policy by using the SAS Client 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 SAS Client 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 128 in this chapter for more information.

For additional assistance, consult your Opware SAS Support Representative or see the *Opware[®] SAS Planning 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 SAS Web Client includes permissions by customer, facility, and User Group Types.

See “User and Group Setup” on page 71 in Chapter 2 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 SAS Web Client 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 SAS Web Client generated and cached the data. Click **Refresh** to refresh that cached data.

Opware administrators can also use the System Diagnosis tools in the SAS Web Client to view information about the health of the multimaster components.

See “Opware SAS Diagnosis” on page 159 in Chapter 5 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 SAS Web Client 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 SAS Web Client in the London facility, John creates Node A1 as a subordinate node of Node A.
- 2** From the v 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 SAS Web Client by using the Multimaster Tools. The Multimaster Tools are available in all SAS Web Clients.



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 SAS Web Client of a single facility. Do not attempt to resolve the same conflict multiple times from the SAS Web Client 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 3-2.

Figure 3-2: Transaction Table That Shows Conflicts

Multimaster Tools : State View				SOURCE FACILITY	
State View		Conflict View			
Refresh					
Key					
Problem		Potential Problem		Good	
■ Conflict		■ Not Sent		■ Not Received	
				□ Unable To Connect	
				■ Received	
Transaction Status Counts					
		C33		C34	
D E S T I N A T I O N	C33			■ 5 ■ 1	
	C34	■ 5 ■ 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 3-3, is listed at the top of the page.

Figure 3-3: 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.

- 2 To resolve a conflict, select the Conflict View tab. The Multimaster Tools: Conflict View page appears, as shown in Figure 3-4.

Figure 3-4: 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 SAS Web Client 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 114 in this chapter for more information.

- 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 3-5.

Figure 3-5: 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 ZMy2clBBE2pN2LdXSiKb0GgkwdqG2 VbM7klQ R awy s6T X oXPvRpRjqw HdRGgJPLg Bh CP7sSGJfS1	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	78vzxGYnqjCbgqjD StA1VU3LZkBSyY4 M NRJfPPRZyL WxVaiNAr P0otheHMnLHMRA nX Ih 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 mwJWim8xPHM9nB0u mGOGX0 HPgQB4438SzTrguhK2P11w A49w2 JE7QG99vuiznC rwC1ysjeB P sXsWrtZ8dx	OJzpb6C K FXueIN8PcJg3KFe7 juKlaqTIVoTAEMdtiV0sA1Ew4ZPAwV 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"/>

- 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 3-6.

Figure 3-6: 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 3-7.

Figure 3-7: 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 ZMv2cIBBe2pN2LdXSikB0GqKwdqG2 Vbm7kIQ R aWY s6T X olXPvRpRjqw HdRGgJPLg Bh CP7sSGgJfS1	SZXT3Aip fKk ZMv2cIBBe2pN2LdXSikB0GqKwdqG2 Vbm7kIQ R aWY s6T X olXPvRpRjqw HdRGgJPLg Bh CP7sSGgJfS1
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	78vzxGYnqICbgqfjD StA1VU3LZkBSyY4 M NRJfPPRZyL WxValNAr POOtheHMnLHMRA nX Ih J 1kQIK zMLr8l Yh YrFi	QuuM YPFNFH2cT 0wspWxvPZDGL9doTSvm9L8F z FtZ8yQPdW7Es 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 FXuelN8PcJg3KFe7 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 3-8 shows.

Figure 3-8: 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 VV85QxGj6k W zL eqic	rWC1ysjeB P sXsWrtZ8dxZY10QvHR3KaQxGSWcG0IPqz 0CCgE7I31tgKA5rAftyPrZX LJChwR VV85QxGj6k 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
Mark Resolved		

- Click **Mark Resolved**. The Multimaster Tools: Mark Conflict Resolved page appears, as Figure 3-9 shows. The page displays the results of marking a transaction resolved.

Figure 3-9: 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 3-10.

Figure 3-10: 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 SAS Web Client 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 3-11.

Figure 3-11: 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 U7vxdx 22 XBz0R bbYYN LhbkhwwlbjZHPsyM4yqWwRQWIZMIE09GLvqTZQoaVctOg5w qj 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 Fz8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu LxKq Q5tk8E1NE n iY97Nk GsrRVzlrC9vltG7O N	jnlpeQuuM YPFNFH2cT 0wspWwvPZDGL9doTSym9L8F z Fz8yQPdW7Es 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		
DB Field	C34	C33
CHANGE_SUMMARY		
CONFLICTING		
DVC_CHANGE_LOG_ID		
DVC_ID		
MODIFIED_BY		
MODIFIED_DT		
TRAN_ID		
	<input type="button" value="Synchronize From"/>	<input type="button" value="Synchronize From"/>

- 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 3-12 shows.

Figure 3-12: 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 114 in this chapter for more information

- Click **Update** beside the drop-down list. The Multimaster Tools: Transaction Synchronization Results page appears, as shown in Figure 3-13.

Figure 3-13: 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.

- Click **Mark Resolved**. The Multimaster Tools: Mark Conflict Resolved page appears. The page displays the results of marking a transaction resolved.
- 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.
- SAS Web Clients cannot contact the multimaster central Data Access Engine.

Production experience for multimaster configurations supports the performance data that Table 3-1 shows.

Table 3-1: 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 SAS Web Clients 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 “Opware SAS Configuration” on page 241 in Chapter 7 for more information.

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 Opware SAS Support Representative for assistance troubleshooting and resolving Opware SAS problems that affect the multimaster operation.

See Table 3-2 for error messages.

Table 3-2: 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	Opware SAS Problem	No value was specified for a given configuration parameter. Log into the SAS Web Client and provide the value for this configuration parameter. Contact your Opware SAS Support Representative for assistance setting Opware 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 3-2: Multimaster Error Messages (continued)

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 3-2: Multimaster Error Messages (continued)

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 Opware SAS in multimaster mode.</p> <p>Run the Opware Admin Multimaster Tools and verify that the transaction data does not contain special characters SOAP might be unable to interpret.</p>
vault.TibcoError	Opware 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 Opware SAS in multimaster mode.</p> <p>Resolve the TIBCO transport error. See <i>TIBCO User's Guide</i>. Restart the Model Repository Multimaster Component.</p>
vault.UnknownError	Opware SAS Problem	<p>The Model Repository Multimaster Component process encountered an unknown error. Contact technical support and provide the database name and Opware SAS component's log file.</p>

Chapter 4: 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 Planning and Installation Guide*.

In Figure 4-1, a Satellite is linked to a standalone core via the Gateway and in Figure 4-2, two Satellites are linked to an Opsware core via the Gateway.

Figure 4-1: A Standalone Core with a Single Satellite

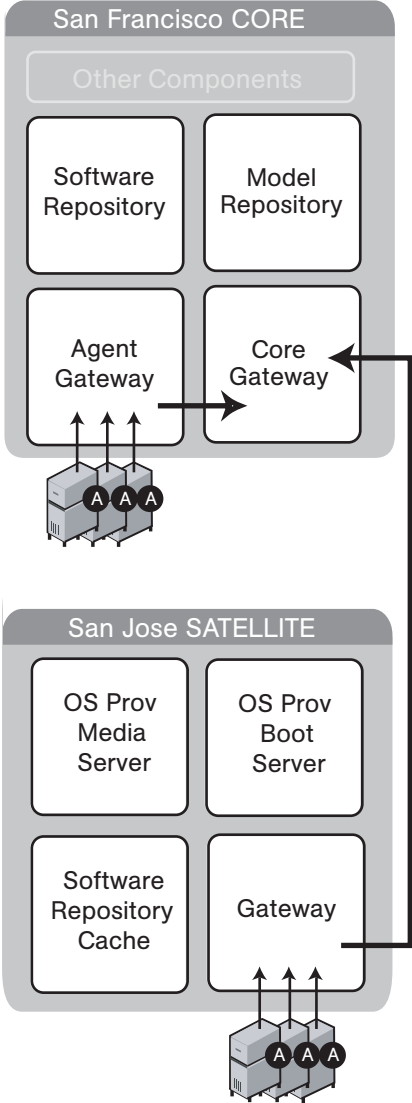
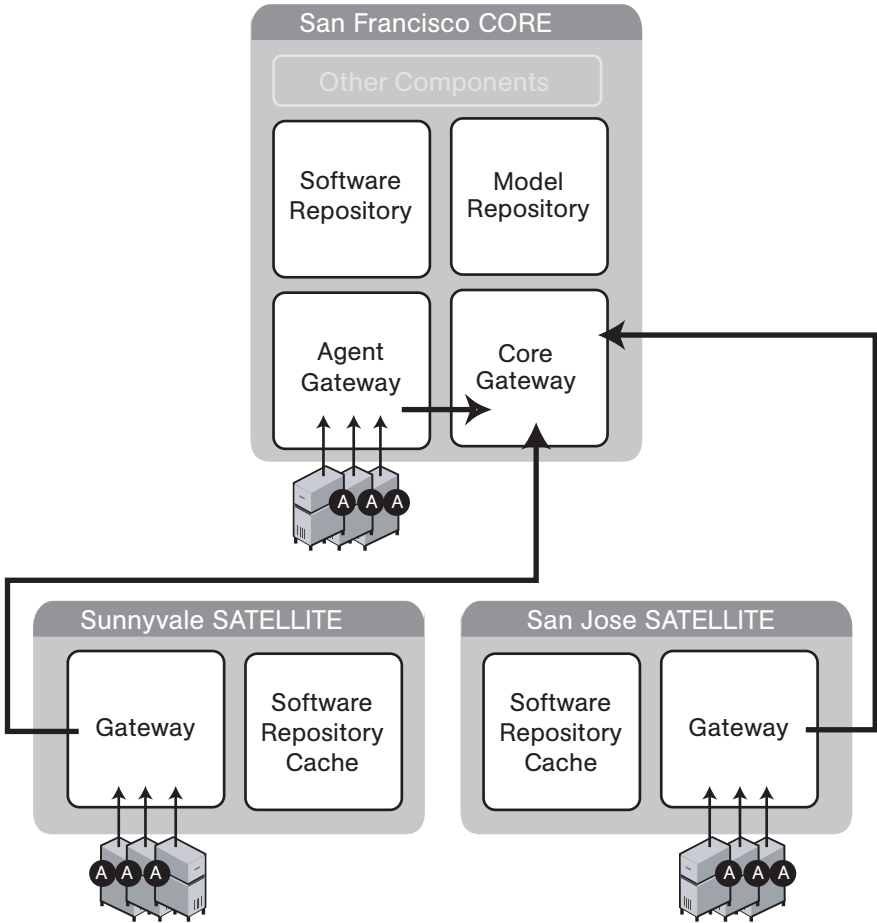


Figure 4-2: Standalone Core with Multiple Satellites



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 "User and Group Setup" on page 71 in Chapter 2 for more information about user groups and Opware permissions.

Viewing Facilities

The Facilities page in the SAS Web Client lists the core and Satellite facilities. In particular, the Facilities page displays Unreachable Facilities, as shown in Figure 4-3.

Figure 4-3: 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 4-4.

Figure 4-4: Realms in Facilities

Facilities: Realms for "GREEN"

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 4-5.

Figure 4-5: Satellite Configuration Parameters



Enabling the Display of Realm Information

By default, the SAS Web Client does not display realm information, which is needed by users who manage Gateways and Software Repository Caches.

To enable access to the realm information, perform the following steps:

- 1** Log into the SAS Web Client as a user that belongs the Administrators group and to a group that has the Configure Opsware permission.
- 2** From the navigation panel, click Administration ► System Configuration.
- 3** Select the Opsware Server Automation System Web Client link.
- 4** In the System Configuration page, for the name `owm.features.Realms.allow`, type the value `true`.
- 5** Click **Save**.

Viewing the Realm of a Managed Server

When installed in a Satellite configuration, Opsware 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 SAS Web Client displays additional information to make it clear which server contains the server corresponding to the IP address, as shown in Figure 4-6.

Figure 4-6: 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="text" 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)						
Opware Lifecycle:	Managed						
Server ID:	510001						

Viewing Gateway Information

To access the Manage Gateway feature, click Administration ► Gateway in the SAS Web Client navigation panel. The Manage Gateway page appears, as Figure 4-7 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 4-7: Status Page of the Manage Gateway Feature

The screenshot shows the 'Manage Gateway' interface for gateway 'cgw0-C28'. It includes a navigation menu on the left, a status bar at the top, and several data tables. A blue arrow points to the 'cgw0-C28' entry in the left-hand list, labeled 'Gateway Selection'.

Gateway Information:
 Gateway: cgw0-C28 Realm: C28 Root: true Version: 1.8.22/1.5 Uptime: 3:6:11.00

Gateway Status Table:

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
cgw0-C29 (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	26460755	62224516	25523838	48682818	3:6:5:25.80	127.0.0.1:50991

Tunnel Mgmt Table:

[TunnelMgmt]	[HighPriority]	[Local]	cgw0-C29 (8192)
0/22/128	0/2/1024	0/2/2048	0/39/1024
0:10:30:57.83	0:2:42:4.43	0:17:43:34.38	3:5:40:20.78

Route Table:

Route	Balance	Resolve	Connect	Discard
0/32/128	0/0/1024	0/1/1024	0/1/1024	0/1/1024
0:10:38:21.82	3:6:6:11.64	0:1:22:24.84	0:1:22:24.84	3:5:40:9.23

MsgProcessor Table:

0/4/1024	0/4/1024	0/4/1024	1/4/1024	0/4/1024	0/4/1024	0/4/1024
2:20:42:20.67	2:20:41:55.24	2:20:42:20.74	2:20:40:34.26	2:20:40:38.62	2:20:43:18.69	2:20:40:35.56
2:20:42:20.67						

DataMover Queue Table:

Active Queues	Total Queued Packets	Total Queued RAM
0	0	0.00 Bytes

Queue Parameters Table:

TAC	TCG	FAC	PAC	POC	ACC	PCC	UAC	UCC	UOC
2	3	2	1	1	0	6	0	0	0

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 4-8 shows.

Figure 4-8: Manage Gateway Feature – Status Page

The screenshot shows the 'Manage Gateway' interface. On the left is a list of gateways: gw2-nat1, Planet10, Purple11, Swing02, gw2-nat2 (selected), and ThereYouAre. The main area displays details for gw2-nat2: Gateway: gw2-nat2, Realm: mars, Root: false, Version: 1.8.5/1.5, Uptime: 79:2:28. Below this is a menu bar with links: Status, Flows, Routing, PathDB, LSDB, Config, History, Ident, Bandwidth. A table of active tunnels is shown below:

Gateway	Cost	BWLimit Kbits/sec	Send BW Kbits/sec	Recv BW Kbits/sec	Total Bytes
gw1-nat2	1	0	0.00	0.83	6865784

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 Opware 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 Opware 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 Opware core or from another Software Repository Cache, and supplies the cached files to Opware Agents on managed servers. The Opware 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. Opware SAS indicates whether a package is available locally or whether the Satellite needs to obtain an update from the Software Repository in the Opware core.

The SAS Web Client 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 Opware SAS is attempting to remediate the software onto a managed server, the SAS Web Client generates an Opware error and displays a complete list of missing packages to help you identify the packages that need to be staged.



The SAS Web Client 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 `/opt/opware/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):

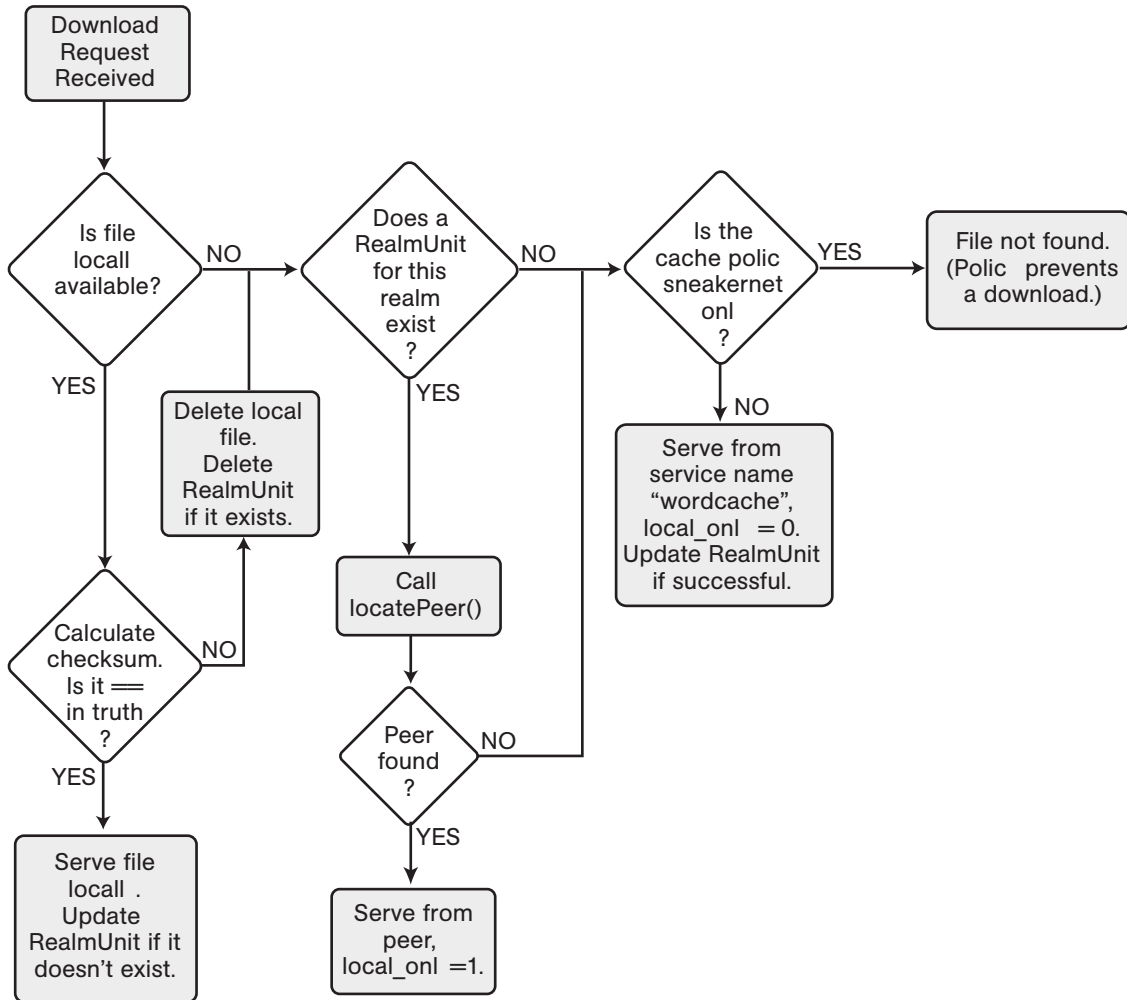
- **On-demand Update:** The local Software Repository Cache obtains current files when needed from the Software Repository in the Opware core.
- **Manual Update:** Software packages are staged to a Satellite's Software Repository Cache in advance of package installation so that performance will be about 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 155 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 4-9 illustrates the logic that the Software Repository Cache uses to update packages in a Satellite.

Figure 4-9: 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 SAS Web Client 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 4-10. In the SAS Web Client, On-demand Update is referred to as Just-in-time (JIT) and Manual Update is referred to as Sneakernet.

Figure 4-10: 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.

Name	Value
osprov.stage2_host: null	<input checked="" type="radio"/> Use default value: buildmgr <input type="radio"/> Use value: <input type="text" value=""/>  
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 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 make take up to five minutes for your change to take affect.

On-demand Updates

Each time an Opware 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 Opware Agent.

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



For security purposes, Opware 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 Opware 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 Opware 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 SAS Web Client 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 policies.

See the *Opware® SAS Content Utilities 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 SAS Web Client component in the Opware core, copy the following files from the `/var/opt/opware/crypto/occ` directory:

```
opware-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 SAS Web Client. If you want to export packages from some other category of software in the SAS Web Client, you need to create a different filter. See the *Opsware® SAS Content Utilities 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 /opt/opsware/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 Opsware 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/opt/opsware/crypto/gateway/token.srv`.
- 2** Log into the server running Opsware Software Repository component.
- 3** Enter the following command to change directories:

```
cd /opt/opsware/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, Opware 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 155 in this chapter for information about how to stage files.

Chapter 5: Opsware SAS Maintenance

IN THIS CHAPTER

This section discusses the following topics:

- Possible Opsware SAS Problems
- Opsware SAS Diagnosis
- The Health Check Monitor for an Opsware SAS Core
- Extensibility of the Health Check Monitor
- Logs for Opsware Components
- Global Shell Audit Logs
- Start Script for Opsware SAS
- Opsware Software
- Mass Deletion of Backup Files
- Designations for Multiple Data Access Engines
- Web Services Data Access Engine Configuration Parameters
- Opsware Global File System Administration

Possible Opsware SAS Problems

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

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

While maintaining Opsware 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 Opsware component, which causes other components to fail

The following examples describe the effects of some component failures:

- If the Data Access Engine fails, the SAS Web Client 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 User's Guide: Server Automation* 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 159 in this chapter for more information.
- Reviewing error logs for Opsware components. See “Logs for Opsware Components” on page 178 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:

- **Standalone 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 Standalone 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 Standalone Tests
- Opsware Agent Comprehensive Tests
- Component Standalone 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: Standalone and comprehensive.

Standalone 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.

Errors Caused By Additional Database Privileges

If an additional privilege (permission) has been made manually to the Oracle database (Model Repository), the following error message might appear:

```
Test Results: The following tables differ between the Data
Access Engine and the Model Repository: facilities.
```

To fix this problem, revoke the database grant. For instructions, see “Troubleshooting System Diagnosis Errors” in the *Opware® SAS Planning and Installation Guide*.

Software Repository Tests

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

Standalone 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: Standalone and comprehensive.

Standalone 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.

Standalone 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.

Standalone 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, your user must belong to the Administrators group. The SAS Web Client 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 5-1.

Figure 5-1: System Diagnosis Page That Shows Opware Components Selected for Testing on the Indicated Facility

System Diagnosis: Perform Diagnosis

Facility: ▼

Specify Diagnosis Options

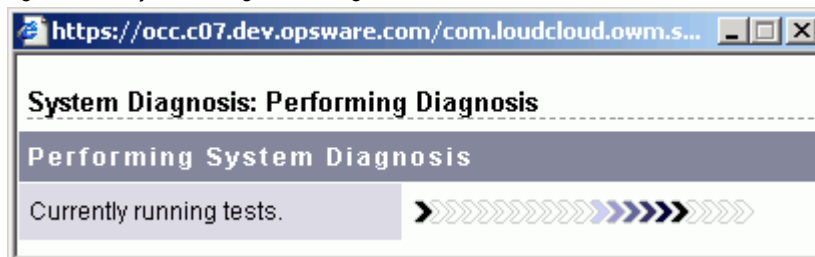
Select the Opware Components you would like to test in the selected datacenter.

Opware Components:	<input checked="" type="checkbox"/> Data Access Engine <input checked="" type="checkbox"/> Software Repository <input checked="" type="checkbox"/> Command Engine <input checked="" type="checkbox"/> Model Repository, Multimaster Component <input checked="" type="checkbox"/> 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 5-2 shows.

Figure 5-2: System Diagnosis Progress Bar



When all the tests are complete, the window closes and the System Diagnosis: Failed Tests page appears in the main SAS Web Client 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.

The Health Check Monitor for an Opware SAS Core

The Health Check Monitor (HCM) includes a suite of tests to check the state of an Opware SAS core. You can run the HCM on Opware SAS version 6.0.2 or greater. The scripts that comprise the HCM are installed by the Opware Installer. There is some overlap between the HCM documented in this section and the System Diagnosis Tool described in “Opware SAS Diagnosis” on page 159.

The HCM has two types of tests:

- **Local Tests:** Validate the health of a core on a component-by-component basis.
- **Global Tests:** Validate the health of a core on a holistic basis.

Overview of HCM Local Tests

The HCM local tests validate individual core components. The local tests reside on the same server as the components they validate. You run local tests by running the Opware SAS Start script (`/etc/init.d/opware-sas`) and specifying the test modes and optional component names. The mode determines which set of tests to run. (You cannot specify individual tests.) Each test is run only once, even if you specify multiple components that require the same test. The test results are displayed on `stdout`.

Validating Core Components by Running HCM Local Tests

To run the local tests, perform the following steps:

- 1 Log on as `root` to the server running the Opware SAS components that you want to check.
- 2 Enter the `opware-sas` command, specifying the mode (category of tests) and one or more components. (See the next section for details on the command options.) For example, the following command verifies that the Web Services Data Access Engine (`twist`) is available:

```
/etc/init.d/opware-sas status twist
```

Syntax of the Script for HCM Local Tests

The script that runs HCM local tests has the following syntax:

```
/etc/init.d/opware-sas <mode> [<component> [<component>...]]
[<name>=<value> [<name>=<value>] ...]
```

Table 5-1 describes the options for this syntax. For a description of the `opware-sas` options for starting and stopping a core, see Table 5-6.

Table 5-1: Options for the HCM Local Test Script

OPTION	DESCRIPTION
mode	<p>The category of tests to run. The mode can be one of the following strings:</p> <ul style="list-style-type: none"> • <code>status</code>: Runs tests that verify the availability of the specified components. For example, the tests verify that the components are listening on the correct ports and responding to basic queries. • <code>verify_post</code>: Same as <code>status</code>. • <code>verify_pre</code>: Runs tests that validate the conditions necessary for the specified components to operate. • <code>verify_functionality</code>: Runs tests that are similar to the tests run by the <code>status</code> mode; however, they might take longer to run. Therefore, you might choose to skip these tests to save time. • <code>health</code>: Runs the tests of the <code>status</code>, <code>verify_pre</code>, and <code>verify_functionality</code> modes and provides an overview of the overall state of the specified components.
component	<p>The internal name of the core component. If this option is not specified, then all components are validated. To view the internal names of the components installed on the local server, enter the following command:</p> <pre>/etc/init.d/opware-sas list</pre>

Table 5-1: Options for the HCM Local Test Script (continued)

OPTION	DESCRIPTION
name=value	<p>Options that control how the tests are run. Allowed values:</p> <ul style="list-style-type: none"> • <code>terse=[true false]</code>: If <code>true</code>, summarizes the results of all successful tests for each component in a single SUCCESS message; however, the results of failed tests are displayed individually. By default, this option is set to <code>false</code>. (This option is passed to the individual tests.) • <code>parsable=[true false]</code>: If <code>true</code>, summarizes the results from all tests for each component with a single SUCCESS or FAILURE message. By default, this option is set to <code>false</code>. (This option is passed to the individual tests.) • <code>verify_filter=<regex></code>: Runs only the tests whose file names match the regular expression you enter. For example, specifying <code>verify_filter="OPSW"</code> runs only tests with file names that contain the string OPSW, such as <code>100_OPSPcheck_host_spin.sh</code>. By default, this option is not defined. (This option is not passed to the individual tests.) <p>If a given test is a symbolic link to another file, the filter will be evaluated against the target of the symbolic link, not the name of the symbolic link. If the test is a symbolic link, <code>verify_filter</code> uses the file name of the file it is pointing to for comparisons.</p>

Overview of HCM Global Tests

The HCM global tests check the entire Opware SAS core. You run these tests by executing the `run_all_probes.sh` script on the server running the Model Repository. Test results are displayed on `stdout`. The global tests cannot check the status of other cores in a multimaster mesh.

In a multi-server core, the global tests running on the Model Repository server connect to the other core servers with SSH. All connections are made as `root`. Authentication is performed by specifying the `root` password or the key file on the command line. If both are specified, then the `root` password is used. One of these authentication methods must be specified unless the server is the local host.

Validating a Core By Running HCM Global Tests

To run the HCM global tests, perform the following steps:

- 1** Log in as `root` to the server running the Model Repository component in the core.
- 2** Execute the `run_all_probes.sh` script with the `run` option. (See the following section for details on the options.) For example, to check the tablespace usage in the Oracle database of the Model Repository, enter the following command:

```
/opt/opware/oi_util/bin/run_all_probes.sh run \
check_database_tables
```

Syntax of the Script for HCM Global Tests

The script that runs HCM global tests has the following syntax:

```
/opt/opware/oi_util/bin/run_all_probes.sh run|list
[<test> [<test>...]]
[hosts="<system>[:<password>] [<system>[:<password>]]..."]
[keyfile=<keyfiletype>:<keyfile>[:<passphrase>]]
```

Table 5-2 describes the options for this syntax.

Table 5-2: Options for the HCM Global Test Script

OPTION	DESCRIPTION
<code>list</code>	Lists the available tests.
<code>run</code>	Runs the specified tests.

Table 5-2: Options for the HCM Global Test Script (continued)

OPTION	DESCRIPTION
test	<p>The name of the test to run. If no tests are specified, all tests are run. When shipped, the script includes the following tests:</p> <ul style="list-style-type: none"> • <code>check_opsware_services</code>: Runs the local tests on all specified servers by running the following command remotely on each core server: <code>/etc/init.d/opsware-sas health</code> • <code>check_MM_state</code>: For a multimaster source core, checks the multimaster state of the core. • <code>check_time</code>: In a multi-server core, verifies that the system clocks are in sync across core servers. • <code>check_opsware_version</code>: Validates that the versions of all the components in the core are the same version. • <code>check_database_tables</code>: Validates that the Model Repository tablespace usage is within acceptable limits. For more information on tablespaces, see "Oracle Setup for Model Repository" in the <i>Opware® SAS Planning and Installation Guide</i>. • <code>check_OS_resources</code>: Validates whether the virtual memory and disk space on Opware SAS partitions is within acceptable thresholds.
system:password	Specifies a remote Opware SAS core server (host name or IP address) and optional <code>root</code> password for the server.
keyfiletype	<p>Specifies the type of key file to use. Allowed values:</p> <ul style="list-style-type: none"> • <code>rsa_key_file</code> • <code>dsa_key_file</code>.
keyfile	Specifies the file containing the current server's SSH private key.
passphrase	Specifies the passphrase that was used to encrypt the SSH private key.

Setting up Passwordless SSH for Global Tests

The global tests access remote servers in a core through the SSH daemon. These tests require you to supply `root` passwords or to use SSH public/private keys.

To set up authentication using public/private keys generated by `ssh-keygen`, perform the following steps:

- 1 Run the following commands on the trusted server and accept the defaults. The commands are different for Linux and Solaris.

Linux:

```
cd /root/.ssh
ssh-keygen -t dsa
```

Solaris:

```
cd /.ssh
ssh-keygen -t dsa
```

- 2 Update the client server by copying the `id_dsa.pub` file to the client server's `.ssh` directory and then renaming it to `authorized_keys`. Here are some example commands for Linux and Solaris:

Linux:

```
scp id_dsa.pub <host>:/.ssh/authorized_keys
/root/.ssh/authorized_keys
```

Solaris:

```
scp id_dsa.pub <host>:/.ssh/authorized_keys
/.ssh/authorized_keys
```

- 3 Verify the trusted server. Run the following command to validate that the trusted server can connect to the client server without a password:

```
ssh -l root <host>
```

Extensibility of the Health Check Monitor

This section is intended for advanced system administrators with experience in Unix shell programming and Opware SAS administration.

The Health Check Monitor (HCM) is implemented as a series of Unix shell scripts that perform local or global tests on the core servers. The scripts conform to specific naming conventions and reside in pre-defined directories. You can extend the HCM by writing your own scripts and copying them to the correct directories under `/opt/opsware/oi_util`.

Requirements for Extensions to HCM Local Tests

An HCM local test is a script that is run by the `/etc/init.d/opsware-sas` script. (See “Validating Core Components by Running HCM Local Tests” on page 166.) A local test script must meet the following requirements:

- **Unix Shell Script:** It is a Unix shell script that runs as `root`.
- **Component Server:** The script resides and runs on the server of the component validated by the script. For example, if the script validates the Data Access Engine (`spin`), it resides on the server that runs the Data Access Engine.
- **Executable:** The script is an executable file (`chmod u+x`).
- **File Name:** The file name of the script has the following syntax:

```
<int><test>.sh
```

In this syntax, `int` is an integer that specifies the test execution order and `test` is the name of the test. Note that the HCM scripts provided with Opware SAS contain `OPSW` in the script file name; for example, `100_OPSPWportping.sh`.

- **Directory:** The script resides in the following directory:

```
/opt/opsware/oi_util/local_probes/<component>/[verify_pre |  
verify_post | verify_functionality]/
```

In this path, `component` is the internal name of the core component, such as `spin` or `twist`. The directories beneath the `component` directory match the category of the test. For example, if the test performs a runtime validation on a core component, the script resides in the `verify_functionality` subdirectory. For details, see “Categories and Local Test Directories” on page 174.

The directories beneath the `component` directory map to the `mode` options of the `/etc/init.d/opsware-sas` command. For example, if you save a script in the `verify_pre` subdirectory, the script is executed when you run `opsware-sas` with the `verify_pre` option. If you specify the `health` option of `opsware-sas`, the

scripts in all three directories are executed. The following table describes the mapping between the directory names and the mode options.

Table 5-3: Modes of opware-sas and the Subdirectories of Local Test Scripts

MODE OPTION OF COMMAND LINE	SUBDIRECTORY OF SCRIPTS RUN FOR THIS OPTION
health	verify_pre verify_post verify_functionality
status	verify_post
verify_functionality	verify_functionality
verify_post	verify_post
verify_pre	verify_pre

- **Exit Code:** The script returns an exit code of zero to indicate success or non-zero for failure. The `/etc/init.d/opware-sas` command uses the exit code to determine the status for the test.
- **Results Displayed:** The script displays test results on `stdout`.
- **Local Preamble Script:** The test script runs the `local_probe_preamble.sh` script, as shown by “HCM Local Test Example” on page 175. The `local_probe_preamble.sh` script contains a superset of the libraries and shell variables used by the `/etc/init.d/opware-sas` command.

The `local_probe_preamble.sh` script performs the following tasks:

- Sets shell variables used by the local tests. For example, it sets `$PYTHON` (which points to the Python 1.5.2 interpreter) and `$UTILS_DIR` (which points to the directory of utilities available to the tests).
- Parses the command line, evaluates all `name=value` pairs, and sets shell variables. For example, if you specify `timeout=60` on the command line when running `/etc/init.d/opware-sas`, the `local_probe_preamble.sh` script sets the variable `$timeout` to the value 60.
- Provides access to useful functions such as `retry`, which executes a command multiple times until it succeeds or exceeds the specified timeout.

- **Shell Variables:** The test script takes into account the variables specified by the `name=value` options on the command line. For a list of pre-defined names, see the `name=value` option in Table 5-1.

Categories and Local Test Directories

The `/opt/opsware/oi_util` directory has the following subdirectories.

local_probes/<component>/verify_pre

This directory includes prerequisite tests for each component. These tests validate that the necessary conditions exist for the component to operate. For example, the directory `twist/verify_pre` contains the test script `10check_localhost_spin.sh` because the Data Access Engine component must be available for the Web Services Data Access Engine component to function.

local_probes/<component>/verify_post

This directory includes validation tests for each component. These tests verify that a given component is available. For example, the directory `spin/verify_post` contains the test script `10check_primary_spin.sh` to validate that the Data Access Engine component is listening on port 1004 and responds to basic queries.

local_probes/<component>/verify_functionality

This directory includes runtime validation tests for each component. These tests verify that a component is fully operational. They are similar to `verify_post` tests, however, they might take longer to run; therefore, you might choose to skip these tests to save time.

Directory Layout for HCM Local Tests

The following directory layout shows where the local tests reside:

```
/opt/opsware/oi_util/  
|  
|_lib  
| |_local_probe_preamble.sh  
|  
|_local_probes  
| |  
| |_COMMON  
| | |_<test>  
| | |_ ...  
| |  
| |_<component>  
| |
```

```

| | _verify_pre
| | | _ <int><test> (can be symlink to ../../COMMON/<test>)
| | | _ ...
| | _verify_post
| | | _ <int><test> (can be symlink to ../../COMMON/<test>)
| | | _ ...
| | _verify_functionality
| | | _<int><test> (can be symlink to ../../COMMON/<test>)
| | | _...
| | _<component>
| | ...

```

HCM Local Test Example

The following script verifies that the `cron` utility is running on the local server.

```

#!/bin/sh
# Verify that cron is running
# Read in our libraries / standard variable settings and parse
# the command line.
/opt/opware/oi_util/lib/local_probe_preamble.sh
printf "Verify \"cron\" is running:"
process_running=`ps -eo fname | egrep '^cron$' | head -1`
if [ -z "$process_running" ]; then
    echo "FAILURE (cron does not exist in the process table)"
    exit 1
else
    echo "SUCCESS"
    exit 0
fi

```

Requirements for Extensions to HCM Global Tests

An HCM global test is a script invoked by the `run_global_probes.sh` command. (See “Validating a Core By Running HCM Global Tests” on page 169.) A global test script must meet the following requirements:

- **Unix Shell Script:** It is a Unix shell script that runs as `root`.
- **Model Repository Server:** The script resides on the Model Repository Server but it can run remotely on any core server.
- **Executable:** The script is an executable file (`chmod u+x`).
- **File Name:** The file name of the script has the following syntax:

```
<int><test>.sh[.remote]
```

In this syntax, `int` is an integer that specifies the test execution order and `test` is the name of the test specified on the command line. Note that the HCM scripts provided with Opware SAS contain `OPSW` in the script file name; for example, `300_OPSPWcheck_time.sh`.

- **Remote Execution:** If the test script runs on a core server other than the Model Repository server, then the file name must have the `.remote` extension. When you execute `run_all_probes.sh` and specify such a test, the script is automatically copied to all specified servers and executed remotely with the SSH protocol.

The `.remote` file name extension is not required for tests that run on the same server as the Model Repository. Examples of these tests are the checks for Model Repository integrity and multmaster conflicts. If the script does not have the `.remote` extension and it needs to communicate with remote servers, the script must use SSH. The global preamble script includes helper functions for handling remote communications with SSH.

- **Directory:** The script resides in the following directory:

```
/opt/opsware/oi_util/global_probes/[verify_pre | verify_post ]/
```

For details, see “HCM Global Test Directories” on page 178.

- **Exit Code:** The script returns an exit code of zero to indicate success or non-zero for failure. The `run_global_probes.sh` command uses the exit code to determine the status for the test.
- **Results Displayed:** The script displays test results on `stdout`.
- **Global Preamble Script:** The test script runs the `global_probe_preamble.sh` script, as shown by “HCM Global Test Example” on page 177. The `global_probe_preamble.sh` script contains a superset of the libraries and shell variables used by the HCM global tests.

The `global_probe_preamble.sh` script performs the following tasks:

- Sets shell variables used by the tests.
- Parses the command line and evaluates all `name=value` pairs, setting them as shell variables. For example, if you specify `hosts="sys1:pw1 sys2:pw2"` on the command line with `run_all_probes.sh`, the `global_probe_preamble.sh` script sets the variable `$hosts` to the value `"sys1:pw1 sys2:pw2"`.

- Provides access to the following functions:
 - `copy_and_run_on_multiple_hosts`: Copies and executes a shell script on multiple remote servers.
 - `copy_from_remote`: Copies a file from a remote server.
 - `copy_to_remote`: Copies a file to a remote server.
 - `run_on_multiple_hosts`: Runs an existing command on multiple servers.
 - `run_on_single_host`: Runs an existing command on a single server.
- **Shell Variables**: The test script takes into account the shell variables specified by the `name=value` options on the command line.
- **Authentication**: The script sets up authentication or public/private key generation. See “Setting up Passwordless SSH for Global Tests” on page 171.

HCM Global Test Example

The following script checks the free disk space of the file systems used by Opware SAS. This script runs on the core servers specified by the `hosts` option of the `run_all_probes.sh` command.

```
# Check for freespace percentage on Opware SAS filesystems
# Read in our libraries, standard variable settings, and parse
# the command line.
/opt/opware/oi_util/lib/global_probe_preamble.sh
MAX_PERCENTAGE=80
for filesystem in /opt/opware /var/opt/opware \
/var/log/opware; do
# The leading and trailing spaces in the following printf
# are to improve readability.
printf " Checking $filesystem: "
percent_free=`df -k $filesystem 2> /dev/null | \
    grep -v Filesystem | \
    awk '{print $5}' | \
    sed 's/%//'\`
if [ $percent_free -ge $MAX_PERCENTAGE ] ; then
    echo "FAILURE (percent freespace > $MAX_PERCENTAGE)"
    exit_code=1
else
    echo "SUCCESS"
    exit_code=0
fi
done
exit $exit_code
```

Directory Layout for HCM Global Tests:

The following directory layout shows where the global tests reside.

```
/opt/opsware/oi_util/  
|_bin  
| |_run_all_probes.sh  
| |_remote_host.py  
| |_<support_utility>  
| |...  
| |_lib  
| |_global_probe_preamble  
|_global_probes  
| |_verify_pre  
| | |_<int><probe>.remote  
| |_verify_post  
| | |_int<probe> [.remote]  
| |_ ...
```

HCM Global Test Directories

The `/opt/opsware/oi_util` directory has the following subdirectories.

global_probes/verify_pre

This directory includes tests that determine whether the specified servers are core servers. When a global test in this category determines that a server is not running an Opsware SAS component or the server is unreachable, no further tests are run against that server.

Only tests with a `.remote` extension are allowed under the `verify_pre` directory.

global_probes/verify_post

This directory includes tests to determine the state of a specific aspect of the entire core. For example, the directory includes the `600_OPswcheck_OS_resources.sh.remote` script, which checks resources such as virtual memory and disk space.

Logs for Opsware Components

Opsware 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. See also the `syslog.conf` file that was used to configure the Opware Boot Server to determine how the logging has been configured for this component.

Build Manager Logs

These logs are in the following file:

```
/var/log/opsware/buildmgr/buildmgr.log
```

Command Engine Logs

These logs are in the following files:

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

Data Access Engine Logs

These logs are in the following files:

```
/var/log/opsware/spin/spin.err*  
/var/log/opsware/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/log/opsware/samba/log.smbd  
/var/log/opsware/samba/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 Planning and Installation Guide*.

Model Repository Multimaster Component Logs

These logs are in the following files:

```
/var/log/opsware/vault/err*  
/var/log/opsware/vault/log.*  
/var/log/opsware/rvrd/rvrdlog*
```

To configure the log file name, log file size, or logging level, in the SAS Web Client, go to Administration ► System Configuration ► Model Repository Multimaster Component.

Opware Agents Logs

The Agents create the following log files on managed servers.

Unix:

```
/var/log/opsware/agent/agent.log*  
/var/log/opsware/agent/agent.err*
```

Windows:

```
%ProgramFiles%Common Files\opsware\log\agent\agent.log*  
%ProgramFiles%Common Files\opsware\log\agent\agent.err*
```

SAS Web Client Logs

The SAS Web Client does not generate its own logs. The SAS Web Client uses JBoss server, which writes to the following log files:

```
/var/log/opsware/occ/server.log*  
/var/log/opsware/httpsProxy/*log*
```

Software Repository Logs

These logs are in the following files:

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

Software Repository Replicator Logs

These logs are in the following files:

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

Software Repository Multimaster Component Logs

These logs are in the following files:

```
/var/log/opsware/mmword/log*
```

Web Services Data Access Engine Logs

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

```
/var/log/opsware/twist/stdout.log*  
/var/log/opsware/twist/twist.log  
/var/log/opsware/twist/access.log  
/var/log/opsware/twist/server.log*  
/var/log/opsware/twist/boot.log  
/var/log/opsware/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 1log.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.

Opware Gateway Logs

These logs are in the following files:

```
/var/log/opware/gateway-name/opswgw.log*
```

Global File System Server Logs

These logs are in the following files:

```
/var/log/opware/hub/OPSWhub.log*  
/var/log/opware/ogfs/ogsh.err*  
/var/log/opware/adapters/adapters.err*  
/var/log/opware/agentcache/agentcache.log  
/var/log/opware/spoke/spoke-*.log  
/var/log/opware/spoke/stdout.log
```

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
- The commands entered in 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 (rosh)



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 `tail`. For an example that uses the `tail` command, see “Example of Monitoring Global Shell Audit Logs” on page 185.

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/opware/ogfs/mnt/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, Opware 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 Opware user who performed the operation
- Name of the Opware user who performed the operation
- Name of the component that generated the audit event
- Version of the Opware SAS component that generated the audit event
- Name of the Opware 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:

```
jdoue@m185:051202182224813:13   jdoue@m185:051202182224790:12
2006/01/28-12:40:19.622 User.Id=2610003 User.Name=jdoue
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:

```
jdoue@m185:051202182224813:13
```

This ID field has the following syntax:

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

The *n* at the end of the ID field is a sequence number of the audit event generated in a session. 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/opsware/ogfs/mnt/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/opsware/ogfs/mnt/audit/scripts/ogfs-host
```

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

```
23f1d546cc657137fa012f78d0adfd56095c3b5
```

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

Example of Monitoring Global Shell Audit Logs

The following example monitors the commands entered by an end-user who logs into a managed server with a Remote Terminal session.

- 1** In a terminal window, as `root`, log into the core server running the OGFS. The steps that follow refer to this window as the “auditing window.”
- 2** In the auditing window, go to the `audit/event` directory:

```
cd /var/opt/opsware/ogfs/mnt/audit/event/ogfs-host
```
- 3** In the SAS Client, open a Remote Terminal to a Unix managed server.
- 4** In the auditing window, examine the last line in the `audit.log` file:

```
tail -1 audit.log.n
```

For example, the following entry from the `audit.log` file indicates that the Opware user `jdoe` opened a Remote Terminal to the host (Device.Name) `toro.opsware.com`. The event ID is `jdoe@m235:060413184452579:59`.

```
jdoe@m235:060413184452595:60 jdoe@m235:060413184452579:59
2006/04/13-18:44:52.728 User.Id=6220044 User.Name=jdoe
Hub:1.1 GlobalShellAgentLogin 1 OK Device.Id=840044
Device.Name=toro.opsware.com ConnectMethod=JUMP RemotePath=
RemoteUser=root
```

- 5** In the auditing window, go to the `audit/streams` directory:

```
cd /var/opt/opsware/ogfs/mnt/audit/streams/ogfs-host
```
- 6** In the auditing window, use the `tail -f` command to monitor the file that corresponds to the Remote Terminal session. The file name is the same as the event ID. For example, if the event ID is `jdoe@m235:060413184452579:59`, then you would enter the following command:

```
tail -f jdoe*59
```

- 7** In the Remote Terminal window, enter some Unix commands such as `pwd` and `ls`.
- 8** Watch the auditing window. The commands (and their output) from the Remote Terminal session are written to the file in the `audit/streams` directory.

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/opsware/agentproxy/bin/auditverify stream_file_name \  
rsa_key_path
```

Here's an example in bash:

```
STREAMDIR=/var/opt/opsware/ogfs/mnt/audit/streams/acct.opsw.com  
STREAMFILE=jdoe@somehost:051210003000111:61  
RSAKEYPATH=/var/opt/opsware/crypto/waybot/waybot.srv
```

```
/opt/opsware/agentproxy/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/opt/opsware/crypto/agent/agent.srv
```

To change the key file used for signing, modify the `audit.signature.key_path` parameter in the System Configuration page of the SAS Web Client. For instructions on accessing the System Configuration page, see “Configuring the Global Shell Audit Logs” on page 188.

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 SAS Web Client 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 5-4. For instructions on accessing the System Configuration page of the SAS Web Client, see “Configuring the Global Shell Audit Logs” on page 188.

Table 5-4: Parameters for Global Shell Audit Log Configuration

PARAMETER	DESCRIPTION	DEFAULT VALUE
audit.root.dir	The root directory for audit streams and scripts.	/var/opt/opware/ogfs/mnt/audit/
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/opt/opware/crypto/waybot/waybot.srv
audit.stream.archive_days	Audit stream files older than this value (in days) are deleted. 0 means files are never deleted.	10

Table 5-4: Parameters for Global Shell Audit Log Configuration (continued)

PARAMETER	DESCRIPTION	DEFAULT VALUE
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 5-4 on page 187. To configure the parameters, perform the following steps:

- 1** In the SAS Web Client, 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**.

Start Script for Opware SAS

Opware SAS includes a unified Opware SAS Start script. You can use the Start script to display all Opware SAS components installed on a server, to start, stop, or restart all components installed on a server, or to start, stop, or restart specific Opware SAS components.

When running the script on a core server, the Start script performs the necessary prerequisite checks for each component installed on the local system.

When an Opware SAS core consists of components distributed across multiple servers, the Start script does not interact directly with remote servers to start or stop components. However, the Start script can connect to remote servers running Opware SAS components and determine whether prerequisites are met before starting dependent components locally.

When checking prerequisites for components running on remote servers, the Start script uses timeout values to allow for different boot times and speed differences among servers. If any of the prerequisite checks fail, the Start script terminates with an error.

The Start script runs in the background when a server running a component reboots; thus, ensuring that the multiuser boot process will not hang until Opware SAS has fully started.

Dependency Checking by the Start Script

The Start script has knowledge of Opware SAS component dependencies and starts Opware SAS components in the correct order. The prerequisite checks verify that dependencies are met before the Start script starts a given component; thus, ensuring that the Opware SAS components installed across multiple servers start in the correct order.

For example, if the component you are attempting to start requires that another component is running, the Start script can verify whether:

- The required component's hostname is resolvable
- The host on which the required component is running is listening on a given port

Starting the Oracle Database (Model Repository)

Opware SAS stores information in the Model Repository, which is an Oracle database. The Opware SAS Start script does not start the Oracle database, which must be up and running before the Opware SAS components can be started. Before you start the Opware SAS components, be sure to start the Oracle listener and database by entering the following command:

```
/etc/init.d/opware-oracle start
```

Logging by the Start Script

The Start script writes to the following logs:

Table 5-5: Start Script Logging

LOG	NOTES
<code>/var/log/opsware/startup</code>	When the server boots, the Start script logs the full text (all text sent to <code>stdout</code>) of the start process for all Opware SAS components installed on the local system.
<code>stdout</code>	When invoked from the command line, the Start script displays the full text of the start process for the components.
<code>syslog</code>	When the server boots, the Start script runs as a background process and sends status messages to the system event logger.

Syntax of the Start Script

The Opware SAS Start Script has the following syntax:

```
/etc/init.d/opsware-sas [options] [component1] [component2]...
```

When you specify specific components to start, stop, or restart, those components must be installed on the local system and you must enter the names exactly as they are displayed by the `list` option. Table 5-6 lists the options for the Opware SAS Start script. To see the options of the Health Check Monitor (HCM) also invoked with `opsware-sas`, see Table 5-1.

Table 5-6: Options for the Opware SAS Start Script

OPTION	DESCRIPTION
<code>list</code>	Displays all components that are installed on the local system and managed by the Start script. The Start script displays the components in the order that they are started.

Table 5-6: Options for the Opware SAS Start Script (continued) (continued)

OPTION	DESCRIPTION
start	<p>Starts all components installed on the local system in the correct order. When you use the <code>start</code> option to start a specific component, the Start script performs the necessary prerequisite checks, then starts the component.</p> <p>The <code>start</code> option does not start the Oracle database (Model Repository), which must be up and running before the Opware SAS components can be started.</p> <p>Some Opware SAS components, such as the Web Services Data Access Engine (<code>twist</code>), can take longer to start. For these components, you can run the Start script with the <code>start</code> option so that the Start script runs on the local system as a background process and logs errors and failed checks to the component's log file.</p> <hr/> <p>NOTE: When you use the <code>start</code> option to start multiple components installed on a server, the Start script will always run the <code>/etc/init.d/opware-sas</code> command with the <code>startsync</code> option.</p> <hr/>
startsync	<p>The <code>startsync</code> option starts all components installed on the local system in a synchronous mode.</p> <p>When you use the <code>startsync</code> option, the Start script runs in the foreground and displays summary messages of its progress to <code>stdout</code>.</p>
restart	<p>Stops and starts all components installed on the local system in a synchronous mode. First, the Start script stops all local components in reverse order; then, executes the <code>startsync</code> option to restart the components in the correct order.</p>
stop	<p>Stops all components installed on the local system in the correct order.</p> <p>This option does not stop the Oracle database.</p>

Starting an Opware SAS Core

To start a core that has been installed on a single server, perform the following steps:

- 1** Log in as `root` to the core server.
- 2** Start the Oracle listener and database for the Model Repository:

```
/etc/init.d/opware-oracle start
```

- 3** Start all core components:

```
/etc/init.d/opware-sas start
```

Starting a Multiple-Server Opware SAS Core

To start a core that has been installed on multiple servers, perform the following steps:

- 1** Find out which servers contain which Opware SAS core components. To list the components installed on a particular server, log in to the server as `root` and enter the following command:

```
/etc/init.d/opware-sas list
```

- 2** Log in as `root` to the server with the Model Repository and start the Oracle listener and database:

```
/etc/init.d/opware-oracle start
```

- 3** In the order listed in “Details: Start Order for Opware SAS Components” on page 193, log in as `root` to each core server and enter the following command:

```
/etc/init.d/opware-sas start
```

Starting an Opware SAS Core Component

You can specify one or more components to start as long as those components are running on the local system. You must enter the component names exactly as they are displayed by the `list` option of the `opware-sas` command.

To start individual components of an Opware SAS core, perform the following steps:

- 1** Log in as `root` to the server that has the component you want to start.
- 2** (Optional) To list the Opware SAS components installed on a server, enter the following command:

```
/etc/init.d/opware-sas list
```


- 3 Enter the following command, where *component* is the name as displayed by the `list` option:

```
/etc/init.d/opware-sas start component
```

For example, if the `list` option displayed `buildmgr`, you would enter the following command to start the OS Provisioning Build Manager:

```
/etc/init.d/opware-sas start buildmgr
```



Alternatively, you can enter the `startsync` option when starting a component on a server. See Table 5-6 on page 190 in this chapter for a description of the `startsync` option.

Details: Start Order for Opware SAS Components

The Start script starts Opware SAS components in the following order. When stopping an Opware SAS core, the components are stopped in the reverse order.

- 1 `opswgw-cgw0-<facility>`: The Opware core-side Gateway for the facility in which the core is running
- 2 `rvrdscrip`: The RVRD script for TIBCO, which Opware SAS uses as part of its multimaster functionality
- 3 `vaultdaemon`: The Model Repository Multimaster Component
- 4 `dhcpd`: A component of the OS Provisioning feature
- 5 `spin`: The Data Access Engine
- 6 `mm_wordbot`: A component of the Software Repository
- 7 `mm_wordbot-clear`: A component of the Software Repository
- 8 `mmworddaemon`: A component of the Software Repository
- 9 `waybot`: The Command Engine
- 10 `smb`: A component of the OS Provisioning feature
- 11 `twist`: The Web Services Data Access Engine
- 12 `buildmgr`: The OS Provisioning Build Manager
- 13 `opswgw-agw0-<facility>`: The Opware agent-side Gateway for the facility in which the core is running

- 14** opswgw-lb: A component of the Opsware Gateway
- 15** replicator: The Software Repository Replicator
- 16** sshd: A component of the Opsware Global File System Server
- 17** hub: A component of the Opsware Global File System Server
- 18** spoke: A component of the Opsware Global File System Server
- 19** agentcache: A component of the Opsware Global File System Server
- 20** occ.server: A component of the SAS Web Client
- 21** httpsProxy: A component of the SAS Web Client
- 22** opsware-agent: The Opsware Agent

Opsware Software

The Opsware Software function is populated during Opsware SAS installation.

Each component of Opsware SAS is shown by its internal name. You cannot add or delete components or nodes in this area of Opsware SAS.

Table 5-7 shows the internal and external names of Opsware SAS components.

Table 5-7: Opsware Internal and External Component Names

INTERNAL NAME	EXTERNAL NAME
Agent	Opsware Agent
buildmgr	OS Provisioning 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 *Opware[®] SAS Planning 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 SAS Web Client 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: Server Automation*.

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

```
/opt/opware/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 5-8, Configuration File Options, for the arguments, their values, and their descriptions.

Syntax of Backup Deletion Script

```
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 `/opt/opsware/pylibs` 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 5-8 on page 196, 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 5-8: 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)

Table 5-8: Configuration File Options (continued)

ARGUMENTS	VALUES	DESCRIPTION
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.
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.

Table 5-8: Configuration File Options (continued)

ARGUMENTS	VALUES	DESCRIPTION
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=/opt/opware/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 /opt/opware/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 Opware 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 Opware 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 multimaster 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 SAS Web Client as a user that belongs to Opware SAS Administrators group.

The SAS Web Client should be installed and listening. The SAS Web Client home page appears.
- 2** From the navigation panel, click Administration ► Opware Software. 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/opware-sas restart spin
```

Designating the Multimaster Central Data Access Engine

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



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

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

- 1** Log into the SAS Web Client as a user that belongs to the Opware System Administrators group.
- 2** From the navigation panel, click Opware Software under Administration. The Opware 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 | Opware | spin | node.
- 8** Click **Select**.
- 9** Navigate the node hierarchy by clicking each node: Opware | Spin | Multimaster Central.
- 10** Click **Re-Assign**.

- 11 Restart the Multimaster Central Data Access Engine.

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

Web Services Data Access Engine Configuration Parameters

This section discusses how to change these parameters with the SAS Web Client or by editing the configuration file. Be sure to restart the Web Services Data Access Engine after changing the parameters.

Changing a Web Services Data Access Engine Parameter

This section describes how to change the parameters displayed by the SAS Web Client. However, the SAS Web Client does not list all of the Web Services Data Access Engine parameters. If you want to change an unlisted parameter, follow the instructions in the next section.

To change a parameter in the SAS Web Client, perform the following steps:

- 1 Log into the SAS Web Client as a member of the Administrators group (admin) and from the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 2 Under Select a Product, click Web Services Data Access Engine.
- 3 Update the parameters you want to change.
- 4 Click **Save**.
- 5 Restart the Web Services Data Access Engine.

Web Services Data Access Engine Configuration File

The Web Services Data Access Engine configuration file includes properties that affect the server side of the Opware Web Services API 2.2. (These properties are not displayed in the SAS Web Client.) The fully-qualified name of the configuration file follows:

```
/etc/opt/opsware/twist/twist.conf
```



During an upgrade of Opware SAS, the `twist.conf` file is replaced, but the `twistOverrides.conf` file is preserved. When you upgrade to a new version of SAS, to retain the configuration settings, you must edit the `twistOverrides.conf` file. The

properties in `twistOverrides.conf` override those specified in `twist.conf`. The Unix `twist` user must have write access to the `twistOverrides.conf` file.

To change a property defined in the configuration file:

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



You must belong to the Administrators group (`admin`) in order to modify the `twist.conf` file. Once the file is changed, the Web Services Data Access Engine must be restarted to apply the changes.

The following table lists the properties of the configuration file that affect the Opware Web Services API 2.2. 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 Opware SAS objects. This window makes enables software developers to update a client-side cache of objects without having to retrieve all of the objects. For more information, see the API documentation for `EventCacheService`.

Table 5-9: Configuration File for Opware Web Services API 2.2

PROPERTY	DEFAULT	DESCRIPTION
<code>twist.webservices.debug.level</code>	1	An integer value that sets the debug level for the Opware 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.

Table 5-9: Configuration File for Opware Web Services API 2.2 (continued)

PROPERTY	DEFAULT	DESCRIPTION
<code>twist.webservices.locale.country</code>	US	The country Internationalization parameter for the Localizer utility. Currently only the US code is supported.
<code>twist.webservices.locale.language</code>	en	Sets the language Internationalization parameter for the Localizer utility. Currently only the en 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.
<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.

Opware Global File System Administration

This section describes administration tasks for the Opware Global File System (OGFS) that might need to be performed after Opware SAS has been installed.

Additional Option Needed for passwd Command on OGFS Server

This problem occurs if the user needs to change a Unix password on the server that runs the OGFS. Because the Opware PAM module alters `/etc/nsswitch.conf`, the `passwd` program needs an additional option to function correctly.

The `passwd` program would normally be used as in the following example:

```
# passwd username
passwd: Changing password for username
passwd: Unsupported nsswitch entry for "passwd:". Use "-r
repository ".
Unexpected failure. Password file/table unchanged.
```

However, on the OGFS server, the `passwd` program must be used as in the following example:

```
# passwd -r files username
passwd: Changing password for username
New Password:
Re-enter new Password:
passwd: password successfully changed for username
```

Chapter 6: Monitoring Opsware SAS

IN THIS CHAPTER

This section contains the following topics:

- Overview of Opsware SAS Monitoring
- Opsware Agent Monitoring
- Agent Cache Monitoring
- Opsware Command Center Monitoring
- Load Balancing Gateway Monitoring
- Data Access Engine Monitoring
- Web Services Data Access Engine Monitoring
- Command Engine Monitoring
- Software Repository Monitoring
- Model Repository Monitoring
- Model Repository Multimaster Component Monitoring
- TIBCO Monitoring
- Opsware Global File System Server Monitoring
- Spoke Monitoring
- Opsware Gateway Monitoring
- OS Build Manager Monitoring
- OS Boot Server Monitoring
- OS Media Server Monitoring

Overview of Opware SAS Monitoring

Opware Server Automation System (SAS) has a built-in system diagnosis function in the SAS Web Client, which allows you to diagnosis the functionality of the following Opware SAS components:

- Data Access Engine
- Software Repository
- Command Engine
- Web Services Data Access Engine
- Multimaster Infrastructure Components (referred to as the Model Repository Multimaster Component in the Opware SAS documentation)

See the *Opware® SAS Administration Guide* for information on how to use this Opware SAS Web Client feature for these components.

This chapter provides information for performing basic monitoring of the components listed above and for the following additional Opware SAS components:

- Opware Agent
- Agent Cache
- SAS Web Client
- Model Repository
- TIBCO
- Opware Global Filesystem Server
- Spoke
- Opware Gateway
- OS Build Manager
- OS Boot Server
- OS Media Server

The commands and other information shown in this document are identical to what the Opware SAS Web Client does when it runs the System Diagnosis function for the components listed above.

The information contained in this document should be used in situations where the System Diagnosis function cannot be performed because the Opsware SAS Web Client cannot be reached, or if your managed environment is already set up for automated monitoring. In that case, you can use the commands contained in this document to automate your system diagnosis and for monitoring Opsware SAS.

The type of monitoring information described in this document includes:

- Commands to confirm specific component processes are running as well as examples of the expected output
- Commands provided by component and by operating system
- Component specific ports, logs, and administrative URLs



The commands shown in this document must be entered all on one line. However, to make sure that the commands and the resulting output are readable, they might have been modified with spaces, blank lines, and line breaks, or backslashes (\) to indicate where a command has been continued on the following line. Also, the output shown is intended as an example only. The output on your servers will be different.

For a description of each of the Opsware SAS components mentioned in this document, see the “Opsware SAS Architecture” chapter in the *Opsware[®] SAS Planning and Installation Guide*.

Opsware Agent Monitoring

The Opsware Agent is the intelligent agent running on each server managed by Opsware SAS. Whenever a change needs to be made to a managed server, the agent brokers the requests.

For more information about the Opsware Agent, see the “Opsware Agent Management” chapter and the “Opsware Agent Utilities” appendix in the *Opsware[®] SAS User’s Guide: Server Automation*.

To use the Opsware SAS Web Client to test an Opsware SAS core’s communication with an Opsware Agent running on a managed server, see the following sections in the *Opsware[®] SAS User’s Guide: Server Automation*:

- “Agent Reachability Communication Tests” in chapter 4

- Appendix A: Communication Test Troubleshooting

Opware Agent Port

The Opware Agent uses port 1002.

Monitoring Processes for Opware Agents

On **Windows**, from the **Start** menu, choose **Run**. In the Run dialog, enter `taskmgr`. In the Windows Task Manager dialog, click the Process tab and look for the processes called `watchdog.exe` and `python.exe`.

On Unix (Solaris, Linux, AIX, and HP-UX), the Opware Agent has two running processes.

On **Solaris**, execute the command:

```
# ps -flg `awk -F= '($1=="pgrp") {print $2}' /var/opt/
opsware/agent/daemonbot.pid`
```

Running this command should produce output similar to the following output:

```
F S  UID  PID  PPID  C  PRI NI  ADDR  SZ  WCHAN  STIME  TTY  TIME  CMD
      8 S  root 9541 9539  0  41  20  ?    1768 ?    Aug
      08 ?    1:23 /opt
      /opsware/agent/bin/python /opt/opsware/agent/pylibs/
      shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
      agent.args
8 S  root 9539  1  0 99  20  ?    398 ?    Aug 08 ?    0:00 /opt
      /opsware/agent/bin/python /opt/opsware/agent/pylibs/
      shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
      agent.args
```

On **Linux**, execute the command:

```
# ps -flg `awk -F= '($1=="pgrp") {print $2}' /var/opt/
opsware/agent/daemonbot.pid`
```

Running this command should produce output similar to the following output:

```
F S  UID  PID  PPID  C  PRI NI  ADDR  SZ  WCHAN  STIME  TTY  TIME  CMD
1 S  root 2538  1  0 85  0  -    3184 wait4 Sep11 ?    00:00:00
      /opt/opsware/agent/bin/python /opt/opsware/agent/pylibs/
      shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
      agent.args
5 S  root 2539 2538  0  75  0  -    30890 schedu Sep11 ?    00:02:56
      /opt/opsware/agent/bin/python /opt/opsware/agent/pylibs/
      shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
      agent.args
```

The daemon monitor is the process with a PPID of 1. The others are server or monitor threads.

on **AIX**, execute the command:

```
# ps -flg `awk -F= '($1=="pgrp") {print $2}' /var/opt/
opsware/agent/daemonbot.pid`
```

Running this command should produce output similar to the following output:

```
F  S UID  PID  PPID  C  PRI  NI  ADDR  SZ  WCHAN  STIME  TTY  TIME  CMD
40001 A root 110600 168026 0 60 20 2000d018 16208 * Sep 05 - 7:15 /opt/
opsware/agent/bin/python /opt/opsware/agent/pylibs/
shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
agent.args
40001 A root 168026 1 0 60 20 2000f25c 1352 Sep 05 - 0:02 /opt/
opsware/agent/bin/python /opt/opsware/agent/pylibs/
shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
agent.args
```

On **HP-UX**, execute the command:

```
# ps -flg `awk -F= '($1=="pgrp") {print $2}' /var/opt/
opsware/agent/daemonbot.pid`
```

Running this command should produce output similar to the following output:

```
F  S UID  PID  PPID  C  PRI  NI  ADDR  SZ  WCHAN  STIME  TTY  TIME  COMD
1  R root 10009 1 0 152 20 437eb1c0 266 - Sep 22 ? 0:00 /opt/
opsware/agent/bin/python /opt/opsware/agent/pylibs/
shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
agent.args
1  R root 10010 10009 0 152 20 434fb440 2190 - Sep 22 ? 3:29 /opt/
opsware/agent/bin/python /opt/opsware/agent/pylibs/
shadowbot/daemonbot.pyc --conf /etc/opt/opsware/agent/
agent.args
```

Opsware Agent URL

<https://<hostname>:1002>

Opsware Agent Logs

The Opsware Agents create the following log files on managed servers.

Windows:

- %ProgramFiles%Common Files\opsware\log\agent\agent.log*
- %ProgramFiles%Common Files\opsware\log\agent\agent.err*

Unix:

- /var/log/opsware/agent/agent.log*
- /var/log/opsware/agent/agent.err*

Conditions to monitor in the Unix logs:

- Strings containing “Traceback”
- Strings containing “OpwareError”

Agent Cache Monitoring

The Agent Cache is a component that serves Opware Agent installation files during the Opware Agent deployment process. The Agent Cache component caches the most recent version of the Opware Agent that is available. The Opware Discovery and Agent Deployment (ODAD) feature obtains the agent installation binaries from the Agent Cache component during agent deployment.

Agent Cache Ports

The Agent Cache uses port 8081.

Monitoring Processes for the Agent Cache

In all configurations, the Agent Cache component has a single running process.

On **Solaris** or **Linux**, execute the command on the server running the Opware Gateway (in an Opware core and an Opware Satellite):

```
# ps auxwww | grep -v grep | grep agentcache
```

Running this command should produce output similar to the following output:

```
root 22288 0.5 0.1 15920 4464 ? S 19:55 0:08 /opt/opsware/bin/  
python /opt/opsware/agentcache/AgentCache.pyc -d /var/opt/  
opsware/agent_installers -p 8081 -b
```

Agent Cache Logs

The Agent Cache logs are in the following files:

- /var/log/opsware/agentcache/agentcache.log
- /var/log/opsware/agentcache/agentcache.err

Conditions to monitor in the logs:

- Strings containing “Error downloading agent”
- Strings containing “Another process is listening on port”

Opsware Command Center Monitoring

The Opsware Command Center is a web-based user interface to Opsware SAS. In its UI, this component is referred to as the Opsware SAS Web Client.

Opsware SAS users connect to the Opsware Command Center component through an Apache HTTPS Proxy (installed by the Opsware Installer with the Opsware Command Center component).

Opsware Command Center Ports

The HTTPS Proxy uses port 443 (HTTPS) and port 80 and directs connections to the Opsware Command Center component, which uses port 1031 (the Web Services port).

Monitoring Processes for the Opsware Command Center

On **Solaris** or **Linux**, execute the command on the server running the Opsware Command Center component:

```
# ps -eaf | grep -v grep | grep java | grep occ
```

Running this command should produce output similar to the following output:

```
occ 17373 1 6 19:46 ? 00:02:35 /opt/opsware/j2sdk1.4.2_10/bin/
    java -server -Xms256m -Xmx384m -XX:NewRatio=3 -Docc.home=/
    opt/opsware/occ -Docc.cfg.dir=/etc/opt/opsware/occ -
    Dopsware.deploy.urls=,/opt/opsware/occ/deploy/ -
    Djboss.server.name=occ -Djboss.server.home.dir=/opt/
    opsware/occ/occ -Djboss.server.
```



To monitor the Opsware Command Center component, you can also set up an automatic monitoring process to send a URL query (using tools such as Wget) to the Opsware Command Center URL. If the Opsware Command Center component returns its login page, it indicates that both the Apache HTTPS Proxy and Opsware Command Center processes are functioning normally.

Opsware Command Center URL

```
https://occ.<data_center>
```

Opsware Command Center Logs

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

- `/var/log/opsware/occ/server.log*`

- `/var/log/opsware/httpsProxy/*log*`

Conditions to monitor in the logs:

- `java.net.ConnectionException`
- `java.net.SocketException`
- `java.lang.NullPointerException`

Load Balancing Gateway Monitoring

The Load Balancing Gateway provides HA and horizontal scaling in an Opware SAS core. The Load Balancing Gateway allows a core to run multiple Opware Global File System Servers because it load balances incoming SSH connections from the SAS Client to the Opware Global File System Servers. SAS Clients connect to the Load Balancing Gateway and it directs the connection to one of the Opware Global File System Servers.

(When you run the Opware Installer, it installs a Load Balancing Gateway with the Opware Command Center component.)

Load Balancing Gateway Ports

By default, the Opware Load Balancing Gateway uses the port 8080.

Monitoring Processes for the Load Balancing Gateway

In all configurations, the Load Balancing Gateway component has two running process – the Gateway process itself and its watchdog process.

On Solaris or Linux, execute the commands on the server running the Opware Command Center component:

```
# ps -eaf | grep -v grep | grep opswgw | grep lb
```

Running this command should produce output similar to the following output:

```
root 32149 1 0 Sep27 ? 00:00:00 [opswgw-watchdog-2.1.1: lb]
      --PropertiesFile /etc/opt/opsware/opswgw-lb/
      opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
root 32156 32149 0 Sep27 ? 00:24:31 [opswgw-gateway-2.1.1: lb]
      --PropertiesFile /etc/opt/opsware/opswgw-lb/
      opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
      --Child true
```

Load Balancing Gateway Logs

The Load Balancing Gateway logs are in the following files:

- `/var/log/opsware/gateway-name/opswgw.log*`

Conditions to monitor in the logs:

- Strings containing “ERROR”
- Strings containing “FATAL” (indicates that the process will terminate)

Data Access Engine Monitoring

The Data Access Engine simplifies interaction with various clients in Opsware SAS, such as the Opsware Command Center, system data collection, and monitoring agents on servers.

Data Access Engine Port

The Data Access Engine uses port 1004 (HTTPS) externally and 1007 (the loopback interface) for Opsware SAS components installed on the same server.

Multimaster Central Data Access Engine Port Forwarding

SQLnet traffic between the Multimaster Central Data Access Engine in a mesh and the Model Repositories in other Opsware SAS cores in the mesh is routed over the Opsware SAS Gateway mesh.

The `tnsnames.ora` file on the server running the Multimaster Central Data Access Engine points to a specified port on each core-side Gateway in the other SAS cores. The core-side Gateway in the core running the Multimaster Central Data Access Engine forwards the connection to the core-side Gateway in each other core, which in turn forwards it to the Model Repositories in the other cores.

The port number on the core-side Gateway is calculated as $20000 + \text{data_center_id}$. For example, if the multimaster mesh has two facilities, Facility A (facility ID 1) and Facility B (facility ID 2), the Multimaster Central Data Access Engine in Facility A connects to port 20002 on the server running the Gateway to reach the Model Repository in Facility B.

For information about the Multimaster Central Data Access Engine, see “Designations for Multiple Data Access Engines” in the *Opsware[®] SAS Administration Guide*.

For information about the Gateway mesh topology, see “Opsware SAS Topologies” in the *Opsware[®] SAS Planning and Installation Guide*.

Monitoring Processes for the Data Access Engine

On Solaris, execute the command on the server running the Data Access Engine component:

```
# /usr/ucb/ps auxwww | grep -v grep | grep spin | grep -v java
```

Running this command should produce output similar to the following output:

```
root  8010  0.5  0.84541631552  ?  S  19:36:42  4:56  /opt/opsware/bin/python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf /etc/opt/opsware/spin/spin.args
root  8008  0.0  0.1  4040  2080  ?  S  19:36:42  0:00  /opt/opsware/bin/python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf /etc/opt/opsware/spin/spin.args
root  8026  0.0  0.53224018224  ?  S  19:36:57  0:01  /opt/opsware/bin/python /opt/opsware/spin/certgenmain.pyc --start --conf /etc/opt/opsware/spin/spin.args
```

On Solaris, you see multiple process that look like the first line of the output above; however, there should be only one process that contains `certgenmain` in the output.

On Linux, execute the command on the server running the Data Access Engine component:

```
# ps auxwww | grep -v grep | grep spin | grep -v java
```

Running this command should produce output similar to the following output:

```
root  30202  0.0  0.0  13592  1500  ?  S  Sep11  0:01  /opt/opsware/bin/python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf /etc/opt/opsware/spin/spin.args
root  30204  1.3  0.6  154928  25316  ?  S  Sep11  411:15  /opt/opsware/bin/python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf /etc/opt/opsware/spin/spin.args
root  30256  0.1  0.3  28500  13024  ?  S  Sep11  50:35  /opt/opsware/bin/python /opt/opsware/spin/certgenmain.pyc --start --conf /etc/opt/opsware/spin/spin.args
```

Data Access Engine URLs

- `https://spin.<data_center>:1004`

To access the Data Access Engine (spin) UI, you need the browser certificate `browser.p12`.

- `https://spin.<data_center>:1004/ObjectBrowser.py?cls=Account&id=0`

Accessing the second URL fails when the Model Repository component is not running.

- `https://spin.<data_center>:1004/sys/dbstatus.py`

Accessing this URL shows the database connection status in the HTML page. Your automatic monitoring system can use a regular expression to extract the number of active database connections.

Data Access Engine Logs

The Data Access Engine logs are in the following files:

- `/var/log/opsware/spin/spin.err*` (The main Data Access Engine error file)
- `/var/log/opsware/spin/spin.log*` (The main Data Access Engine log file)
- `/var/log/opsware/spin/spin_db.log`
- `/var/log/opsware/spin/daemonbot.out` (Output from the application server)

Conditions to monitor in the `daemonbot.out` log:

- Strings containing “FAILURE”
- Strings containing “ERROR”
- Strings containing “WARNING”
- Strings containing “Traceback”

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

Web Services Data Access Engine Monitoring

The Web Services Data Access Engine provides increased performance to other Opware SAS components.

When you run the Opware Installer, the Web Services Data Access Engine (Twist) component is installed on the server with the Opware Command Center component and on the server with the Opware Global File System Server component.

Web Services Data Access Engine Port

The Web Services Data Access Engine uses port 1032.

The Opware Command Center component and Opware Global File System Server component that are installed on the server with the Web Services Data Access Engine communicate with it on port 1026 (a private loopback port).

Monitoring Processes for the Web Services Data Access Engine

On Solaris, execute the command on the server running the Opware Command Center component and on the server running the Opware Global File System Server component:

```
# /usr/ucb/ps auxwww | grep -v grep | grep \/opt\/opsware\/twist
```

Running this command should produce output similar to the following output:

```
twist  9274  0.0  1.416748054040  ?  S  Aug 08 410:33 /opt/opsware/
        j2sdk1.4.2_10/bin/java -server -Xms16m -Xmx128m -
        Dtwist.port=1026 ..... -classpath opt/opsware/j2sdk1.4.2_
        10/jre .....
twist  9238  0.0  0.1  1088  744  ?  S  Aug 08  0:00 /bin/sh /opt/
        opsware/twist/watchdog.sh start 60
```

On Linux, execute the command on the server running the Opware Command Center component and on the server running the Opware Global File System Server component:

```
# ps auxwww | grep -v grep | grep \/opt\/opsware\/twist
```

Running this command should produce output similar to the following output:

```
twist  4039  0.2  11.3  2058528  458816  ?  S  Sep11  80:51 /opt/opsware/
        j2sdk1.4.2_10/bin/java -server -Xms256m -Xmx1280m -
        XX:MaxPermSize=192m -
        Dorg.apache.commons.logging.Log=org.apache.commons.logging
        .impl.Jdk14Logger .....
twist  4704  0.0  0.0  4236  1124  ?  S  Sep11  1:28 /bin/sh /opt/
        opsware/twist/watchdog.sh start 60'
twist  4743  0.0  0.6  376224  27160  ?  S  Sep11  18:31 /opt/opsware/
        j2sdk1.4.2_10/bin/java -server -Xms16m -Xmx128m -
        Dtwist.port=1026 ..... -classpath /opt/opsware/
        j2sdk1.4.2_10/jre/.....
```


Web Services Data Access Engine URL

`https://occ.<data_center>:1032`

Web Services Data Access Engine Logs

The Web Services Data Access Engine logs are in the following files:

- `/var/log/opsware/twist/stdout.log*`
- `/var/log/opsware/twist/twist.log`
- `/var/log/opsware/twist/access.log`
- `/var/log/opsware/twist/server.log*` (Application level logging)
- `/var/log/opsware/twist/boot.log`
- `/var/log/opsware/twist/watchdog.log`

The `stdout.log` files contain `stdout` and `stderr` and logs the output of any `System.out.println()`, `System.err.println()` and `e.printStackTrace()` messages; however, only some of the exceptions will show up in these logs. The number of files and the size of each file can 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 `stdout.log.5` are progressively older files. The file is also rotated on startup.

The `twist.log` file contains WebLogic-specific messages and WebLogic level exceptions. These files are rotated on startup. Monitor the `twist.log` files for exceptions that indicate when the Web Services Data Access Engine (Twist) component failed to start correctly. If errors are encountered during Model Repository (Truth) connection setup, errors are logged in the `twist.log` files; for example, you might see the following error message:

```
####<Oct 14, 2006 1:37:43 AM UTC> <Error> <JDBC> <localhost.localdomain>
<twist> <main> <<WLS Kernel>> <> <BEA-001150> <Connection Pool
"TruthPool" deployment failed with the following error:
<Specific message, such as Oracle error codes and tracebacks>
```

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` files contain application level exceptions and debug messages generated from the Web Services Data Access Engine. The `server.log` files will also contain errors resulting from Model Repository (Truth) connection setup problems. The debug messages are controlled by the log level set at the package or class level in the

`twist.conf` file. 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.

Command Engine Monitoring

The Opware Command Engine is the means by which distributed programs such as Opware Agents run across many servers. 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.

Command Engine Port

The Command Engine uses port 1018.

Monitoring Processes for the Command Engine

On Solaris, execute the command on the server running the Command Engine component:

```
# /usr/ucb/ps auxwww | egrep '(COMMAND$|waybot)' | grep -v  
grep
```

Running this command should produce output similar to the following output:

```
USER  PID  %CPU  %MEM  SZ  RSS  TT      S  START  TIME  COMMAND  
root 1246   0.0   0.1 4040 2064  ?      S  Sep 24  0:00 /opt/opsware/bin/  
python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf  
/etc/opt/opsware/waybot/waybot.args  
root 1248   0.0   0.41596814592 ?      S  Sep 24  2:19 /opt/opsware/bin/  
python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf  
/etc/opt/opsware/waybot/waybot.args
```

On Solaris, the Command Engine has two processes – one process for the daemon monitor and one process for the server.

On Linux, execute the command on the server running the Command Engine component:

```
# ps auxwww | egrep '(COMMAND$|waybot)' | grep -v grep
```

Running this command should produce output similar to the following output:

```
USER  PID  %CPU  %MEM  VSZ   RSS  TTY  STAT  START  TIME  COMMAND
root  412  0.0   0.0 13600 1472  ?    S     Sep11  0:00  /opt/opsware/
      bin/python /opt/opsware/pylibs/shadowbot/daemonbot.pyc
      --conf /etc/opt/opsware/waybot/waybot.args
```

On Linux servers running kernel 2.4 or later, the Command Engine has one process.

Command Engine URL

`https://way.<data_center>:1018`

Command Engine Logs

The Command Engine logs are in the following files:

- `/var/log/opsware/waybot/waybot.err*`
- `/var/log/opsware/waybot/waybot.log*`
- `/var/log/opsware/waybot/daemonbot.out*`

Software Repository Monitoring

The Software Repository is where all software managed by Opsware SAS is stored.

When the Opsware SAS core is running in multimaster mode, the Software Repository Replicator might be set up for the core. The Replicator provides backup functionality for Software Repositories running in a multimaster mesh to ensure that all packages remain available even when a Software Repository goes offline.

For more information about the Software Repository Replicator, see the “Software Repository Replicator Setup” appendix in the *Opsware[®] SAS Administration Guide* for information.

Software Repository Ports

The Software Repository uses the following ports:

- 1003 (Encrypted)
- 1006 (Cleartext)

- 1005 (Replicator administrative user interface)
- 5679 (Multimaster Software Repository)

Monitoring Processes for the Software Repository

On Solaris, execute the command on the server running the Software Repository component:

```
# /usr/ucb/ps auxwww | grep -v grep | grep mm_wordbot
```

Running this command should produce output similar to the following output:

```

root  8625  0.0  0.1  4048  1912  ?  S  Aug 08  0:00  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot.args
root  8627  0.0  0.52034418600  ?  S  Aug 08  7:38  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot.args
root  8675  0.0  0.1  4032  1904  ?  S  Aug 08  0:00  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot-clear.args
root  8677  0.0  0.210104  8096  ?  S  Aug 08  0:01  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot-clear.args

```

On Solaris, the Software Repository has four running processes – two processes for the encrypted Software Repository and two for the cleartext Software Repository.

On Linux, execute the command on the server running the Software Repository component:

```
# ps auxwww | grep -v grep | grep mm_wordbot
```

Running this command should produce output similar to the following output:

```

root  31006  0.0  0.0  13612  1492  ?  S  Sep11  0:00  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot.args
root  31007  0.0  0.1  103548  7688  ?  S  Sep11  7:33  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot.args
root  31092  0.0  0.0  13608  1480  ?  S  Sep11  0:00  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot-clear.args
root  31093  0.0  0.1  70172  6424  ?  S  Sep11  2:11  /opt/opsware/bin/
        python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
        /etc/opt/opsware/mm_wordbot/mm_wordbot-clear.args

```

On Linux, the Software Repository has multiple running processes (most are threads), which are for the encrypted Software Repository and for the cleartext Software Repository.

In a multimaster configuration, each Software Repository has additional processes.

On Solaris multimaster core, execute these commands on the server running the Software Repository component:

```
# /usr/ucb/ps auxwww | grep -v grep | grep mmword
```

Running this command should produce output similar to the following output:

```
root 10300 0.0 0.1 2784 1680 ? S 19:41:39 0:00 /opt/opsware//
bin/python /opt/opsware//pylibs/shadowbot/etc/
daemonizer.pyc --runpath /var/opt/opsware/mmword/ --cmd
/opt/opsware/j2sdk1.4.2_10/bin/java -ms512m -mx1024m
-DCONF=/etc/opt/opsware/mmword
com.loudcloud.mmword.MMWord
```

```
root 10305 0.0 0.458787215600 ? S 19:41:39 0:00 /opt/opsware/
j2sdk1.4.2_10/bin/java -ms512m -mx1024m -DCONF=/etc/opt/
opsware/mmword com.loudcloud.mmword.MMWord
```

```
# /usr/ucb/ps auxwww | grep -v grep | grep replicator
```

Running this command should produce output similar to the following output:

```
root 10309 0.0 0.1 3152 2056 ? S 19:41:39 0:00 /opt/opsware/
bin/python /opt/opsware/pylibs/shadowbot/daemonbot.pyc
--conf /etc/opt/opsware/replicator/replicator.args
```

```
root 10311 0.0 0.2 6280 5296 ? S 19:41:39 0:00 /opt/opsware/
bin/python /opt/opsware/pylibs/shadowbot/daemonbot.pyc
--conf /etc/opt/opsware/replicator/replicator.args
```

On **Linux multimaster core**, execute the command on the server running the Software Repository component:

```
# ps auxwww | grep -v grep | grep mmword
```

Running this command should produce output similar to the following output:

```
root 29799 0.0 0.0 2292 528 ? S Sep27 0:00 /opt/opsware//bin/
python /opt/opsware//pylibs/shadowbot/etc/daemonizer.pyc
--runpath /var/opt/opsware/mmword/ --cmd /opt/opsware/
j2sdk1.4.2_10/bin/java -ms512m -mx1024m -DCONF=/etc/opt/
opsware/mmword com.loudcloud.mmword.MMWord
```

```
root 29800 0.0 0.0 1245468 1360 ? S Sep27 3:18 /opt/opsware/
j2sdk1.4.2_10/bin/java -ms512m -mx1024m -DCONF=/etc/opt/
opsware/mmword com.loudcloud.mmword.MMWord
```

```
# ps auxwww | grep -v grep | grep replicator
```

Running this command should produce output similar to the following output:

```
root 29806 0.0 0.0 12736 540 ? S Sep27 0:00 /opt/opsware/bin/
python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
/etc/opt/opsware/replicator/replicator.args
```

```
root 29807 0.0 0.0 66684 1108 ? S Sep27 1:08 /opt/opsware/bin/
python /opt/opsware/pylibs/shadowbot/daemonbot.pyc --conf
/etc/opt/opsware/replicator/replicator.args
```

Software Repository URL

`https://theword.<data_center>:1003`

Software Repository Logs

The logs for the Software Repository are in the following files:

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

The logs for the Software Repository Replicator are in the following files:

- `/var/log/opsware/replicator/replicator.err*`
 - Conditions to monitor in the `replicator.err` log: strings containing “[ERROR]”
- `/var/log/opsware/replicator/daemonbot.out`
 - Conditions to monitor in the `daemonbot.out` log:
 - Strings containing “Traceback”
 - Strings containing “Unable to access conf file”
 - Strings containing “Unable to access configuration file”
 - Strings containing “Error: no backticks allowed in the conffile”
- `/var/log/opsware/replicator/replicator.log*`

The logs for the Software Repository Multimaster Component are in the following files:

- `/var/log/opsware/mmword/log*`
 - Conditions to monitor the `log*` files:
 - Strings containing “[ERROR]”
 - Strings containing “Traceback”

Model Repository Monitoring

The Model Repository is an Oracle database that contains essential information necessary to build, operate, and maintain a list of all managed servers, their hardware, their configuration, the operating system and all other applications.

For more information about the Model Repository, including detailed information about monitoring the Model Repository, see the “Oracle Setup for the Model Repository” appendix in the *Opware® SAS Planning and Installation Guide*.

Model Repository Port

The default port for the Model Repository is 1521, however, this might have been modified by the database administrator who installed it.

Monitoring Processes for the Model Repository

Monitor the Oracle Database process. If the process is not found, the database has failed or was not started.

On Solaris or Linux, execute the command on the server running Oracle:

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

Running this command should produce output similar to the following output:

```
oracle    2112      1  0 21:22 ?        00:00:00 ora_pmon_truth
```

(The process name might include the database SID, truth, as shown in this example.)

If the process is not found, the listener has failed or was not started.

On Solaris or Linux, use this command to monitor the Oracle Listener process:

```
# ps -fu oracle | grep tnslnsr
```

Running this command should produce output similar to the following output:

```
oracle    2021      1  0 21:22 ?        00:00:01 /u01/app/oracle/  
product/10.2.0/db_1/bin/tnslnsr LISTENER -inherit
```

Model Repository Logs

Log files for the Model Repository are produced by the Oracle database and their location is specific to your installation.

By default, Opware SAS uses a directory for each SID (in this case truth) for the Model Repository logs. (This could be different based on how Oracle was installed.)

```
/u01/app/oracle/admin/truth/bdump/alter_truth.log
```


Conditions to monitor:

Not all errors indicate a problem with the database. Some errors might be caused by an application.

In these examples, there is a problem if the command has output.

```
grep ORA- /u01/app/oracle/admin/truth/bdump/alter_truth.log
ORA-00600: internal error code, arguments: [729], [480],
[space leak], [], [], [], [], []
ORA-07445: exception encountered: core dump [lxmcpn()+0]
[SIGSEGV] [Address not mapped to object] ...
```

Table Space Usage

Tablespace usage should be monitored against a threshold, usually increasing in severity (for example., over 80% is a warning, over 90% is an error, over 95% is a critical error).

There are several ways to monitor tablespace usage. For a SQL query that you can run to check for sufficient free disk space in the tablespaces, see the “Oracle Setup for the Model Repository” appendix in the *Opsware® SAS Planning and Installation Guide*. The SQL query provided in the installation guide must be executed as a privileged database user.

Multimaster Conflicts

The number of conflicting transactions in any Model Repository can be found by running the following SQL query as any Opsware database user.

```
select count(*) from transaction_conflicts where resolved =
'N' ;
```

Multimaster conflicts should be monitored in stages, with increasing numbers of conflicts resulting in increasing levels of escalation. The values used for the stages depend on patterns of use.

Opsware Inc. recommends that the administrator record the number of conflicts for some period of time (perhaps a week) and use that information to determine the level of alert raised by the monitoring system.

For more information about multimaster conflicts, see the *Opsware® SAS Administration Guide*.

Model Repository Multimaster Component Monitoring

The Model Repository Multimaster Component is a Java program responsible for keeping multiple Model Repositories synchronized and propagating changes for the originating Model Repository to all other Model Repository databases.

Model Repository Multimaster Component Port

The Model Repository Multimaster Component uses port 5678.

Monitoring Processes for the Model Repository Multimaster Component

On Solaris, execute the command on the server running the Model Repository Multimaster component [the server where you installed the Multimaster Infrastructure Components (vault) with the Opware Installer]:

```
# /usr/ucb/ps auxwww | grep -v grep | grep vault | grep -v twist
```

Running this command should produce output similar to the following output:

```
root 3884 0.0 0.1 2792 1568 ? S Jul 26 0:00 /opt/opsware//bin/
python /opt/opsware//pylibs/shadowbot/etc/daemonizer.pyc
--runpath /var/log/opsware/vault --cmd /opt/opsware/
j2sdk1.4.2_10/bin/java -classpath /opt/opsware/vault
..... -ms120m -mx1024m -DCONF=/etc/opt/opsware/vault/
-DHOSTNAME= com.loudcloud.vault.Vault
root 3885 0.0 0.1 1096 848 ? S Jul 26 0:00 /bin/sh -c /opt/
opsware/j2sdk1.4.2_10/bin/java -classpath /opt/opsware/
vault/cl
root 3887 0.0 3.9194192155784 ? S Jul 26 2:34 /opt/opsware/
j2sdk1.4.2_10/bin/java -classpath /opt/opsware/vault
..... -ms120m -mx1024m -DCONF=/etc/opt/opsware/vault/
-DHOSTNAME= com.loudcloud.vault.Vault
```

On Linux, execute the command on the server running the Model Repository Multimaster component [the server where you installed the Multimaster Infrastructure Components (vault) with the Opware Installer]:

```
# ps auxwww | grep -v grep | grep vault | grep -v twist
```

Running this command should produce output similar to the following output:

```
root 28662 0.0 0.0 2284 532 ? S Sep27 0:00 /opt/opsware//bin/
python /opt/opsware//pylibs/shadowbot/etc/daemonizer.pyc
--runpath /var/opt/opsware/vault --cmd /opt/opsware/
j2sdk1.4.2_10/bin/java -classpath /opt/opsware/vault/
classes:/opt/opsware/vault ..... -ms120m -mx1024m
-DCONF=/etc/opt/opsware/vault/
-DHOSTNAME=m234.dev.opsware.com com.loudcloud.vault.Vault
```

```

root 28663 0.0 6.3 1285800 130896 ? S Sep27 5:32 /opt/opsware/
      j2sdk1.4.2_10/bin/java -classpath /opt/opsware/vault/
      classes:/opt/opsware/vault ..... -ms120m -mx1024m
      -DCONF=/etc/opt/opsware/vault/
      -DHOSTNAME=m234.dev.opsware.com com.loudcloud.vault.Vault

```

Model Repository Multimaster Component Logs

The Model Repository Multimaster Component logs are in the following files:

- /var/log/opsware/vault/log.*

Condition to monitor in the logs: The string “Traceback”

To configure the log file name, log file size, or logging level, in the SAS Web Client, go to Administration ► System Configuration ► Model Repository Multimaster Component.

TIBCO Monitoring

In a multimaster mesh, Opware SAS uses the TIBCO Certified Messaging system to synchronize the Model Repositories in different facilities. The Opware Installer automatically installs and configures TIBCO Rendezvous. By default, the installer configures the Rendezvous neighbors in a star topology, in which the source core is at the center.

TIBCO traffic between Opware SAS cores is routed over the Opware SAS Gateway mesh. Each core contains a Model Repository with data that is synchronized with the Model Repositories in other cores. This synchronization data passes through the core Gateways.

For information about this topology, see “Opware SAS Topologies” in the *Opware[®] SAS Planning and Installation Guide*.

For information about TIBCO configuration in an Opware SAS multimaster mesh, see the “TIBCO Rendezvous Configuration for Multimaster” appendix in the *Opware[®] SAS Planning and Installation Guide*.

TIBCO Ports

TIBCO uses the following ports:

- 7500 for connections to the local network (UDP) [TIBCO Rendezvous daemon (rvd)]
- 7501 for neighbor connections [TIBCO Rendezvous routing daemon (rvrd)]
- 7580 for TIBCO Management



The TIBCO Rendezvous routing daemon (rvrd), which listens on port 7501 for neighbor connections from other SAS cores, uses port forwarding through the Opware Gateway mesh to reach TIBCO neighbors in other SAS cores.

The port number on the core-side Gateway is calculated as 10000 + data_center_id_of_the_neighbor. For example, if the multimaster mesh has two facilities, Facility A (facility ID 1) and Facility B (facility ID 2), the core-side Gateway in Facility A forwards port 10002 and the core-side Gateway in Facility B forwards port 10001.

Monitoring Processes for TIBCO

On Solaris, execute the command on the server running the Model Repository component:

```
# /usr/ucb/ps auxwww | grep -v grep | grep rvrd
```

Running this command should produce output similar to the following output:

```
nobody4 3831 0.0 0.52469618000 ? S Jul 26 0:01 /opt/opsware/  
tibco/bin/rvrd -http 7580 -https 7581 -permanent -store  
/var/opt/opsware/rvrd/rvrdstore -logfile /var/log/opsware/  
rvrd/rvrdlog
```

On Linux, execute the command on the server running the Model Repository component:

```
# ps auxwww | grep -v grep | grep rvrd
```

Running this command should produce output similar to the following output:

```
65534 18095 0.0 0.5 162032 10468 ? S 16:39 0:04 /opt/opsware/  
tibco/bin/rvrd -http 7580 -https 7581 -permanent -store  
/var/opt/opsware/rvrd/rvrdstore -logfile /var/log/opsware/  
rvrd/rvrdlog
```

Additionally, on the server running the Software Repository component, execute the commands to monitor TIBCO processes.

On Solaris, execute the command on the server running the Software Repository component:

```
# /usr/ucb/ps auxwww | grep -v grep | grep rvd
```

Running this command should produce output similar to the following output:

```
nobody4 1274 0.0 0.62657610576 ? S Oct 13 0:35 rvd -listen  
tcp:7500 -no-permanent
```

On Linux, execute the command on the server running the Software Repository component:

```
# ps auxwww | grep -v grep | grep rvd
```

Running this command should produce output similar to the following output:

```
65534 8609 0.0 0.3 135024 13708 ? S Oct12 0:29 rvd -listen
      tcp:7500 -no-permanent
```

When the Web Services Data Access Engine (Twist) component is installed on a server without the Model Repository Multimaster Component (vault), you must monitor the TIBCO rvd processes used by the Web Services Data Access Engine (Twist) component. Within an Opware SAS core, TIBCO rvd is used to communicate with the Web Services Data Access Engine (Twist) running on the other servers and notify it to refresh the data in its cache after multimaster transactions are published to the Model Repository in the core.

On Solaris, execute the commands on the server running the Opware Command Center (OCC) component and on the server running the Opware Global File System Server component (aka Hub):

```
# /usr/ucb/ps auxwww | grep -v grep | grep rvd
```

Running this command should produce output similar to the following output:

```
nobody4 1274 0.0 0.62657610576 ? S Oct 13 0:34 rvd -listen
      tcp:7500 -no-permanent
```

On Linux, execute the commands on the server running the Opware Command Center (OCC) component and on the server running the Opware Global File System Server component (aka Hub):

```
# ps auxwww | grep -v grep | grep rvd
```

Running this command should produce output similar to the following output:

```
twist 3500 0.0 0.8 137584 17172 ? S Oct13 1:47 rvd -listen
      tcp:7500 -no-permanent
```

TIBCO URL

`http://truth.<hostname>:7580` (Displays the TIBCO Rendezvous web client)

Where the `<hostname>` is the IP address or fully-qualified host name of the server running the Model Repository Multimaster Component (vault). The TIBCO Rendezvous General Information page appears.

Additionally, you can use the TIBCO Rendezvous web client to check status, such as whether a TIBCO Rendezvous Neighbor has connections to an Opware facility.

TIBCO Logs

The TIBCO logs are in the following file:

- `/var/log/opsware/rvrd/rvrdlog`

(You can adjust the logging level by using the TIBCO Rendezvous web client.)

Opware Global File System Server Monitoring

The Opware Global Shell feature runs on the Global File System Server, which dynamically constructs a virtual file system – the Opware Global File System (OGFS).

The Global File System Server can connect to an Opware Agent to open a Unix shell or a Windows Remote Desktop connection on a managed server.

For information about using the Global Shell, see the Global Shell chapter and appendices in the *Opware® SAS User's Guide: Server Automation*.

The Global File System Server component consists of the following programs:

- **Hub:** A Java program that interacts with other Opware core components and Opware Agents on managed servers (through the Agent Proxy) to compose the filesystem view.
- **Adapter:** On Linux, a C program that transports filesystem requests and replies between the FUSE module in the kernel and the Hub and uses the FUSE userspace library to communicate with the FUSE kernel module. On Solaris, a Python program that communicates with a custom kernel module.
- **Agent Proxy:** A Python program that provides the Hub with SSL connectivity to Opware Agents running on managed servers
- **FUSE (Linux Only):** Filesystem in Userspace (FUSE) (software governed by the GNU GPL license) that provides in-kernel dispatch of filesystem requests into the Adapter

The process group ID file for the Hub is located in the following directory:

- `/var/opt/opsware/hub/hub.pgrp`

All Opware Global File System Server programs (Hub, Adapter, Agent Proxy, and their log rotators) run in this process group.

Monitoring Process for the Opsware Global File System Server

On Solaris, execute the command on the server running the Opsware Global File System Server component:

```
# ptree $(ps -g $(cat /var/opt/opsware/hub/hub.pgrp) -o
pid=)
```

Running this command should produce output similar to the following output:

```
7594 /opt/opsware/bin/python /opt/opsware/hub/bin/rotator.py /opt/
    opsware/j2sdk1.4.2.....
7598 /opt/opsware/j2sdk1.4.2_10/bin/java -server -Xms64m -Xmx1024m
    -Dhub.kernel=SunO.....
7613 /opt/opsware/bin/python /opt/opsware/adapter/SunOS/bin/rotator.py
    /opt/opsware/.....
7617 /opt/opsware/ogfsutils/bin/python2.4 /opt/opsware/adapter/
    SunOS/lib/adapter.py.....
7618 /opt/opsware/adapter/SunOS/bin/mount -o hostpath=
    /hostpath,nosuid /dev/ogdrv /v.....
7619 /opt/opsware/bin/python /opt/opsware/agentproxy/bin/rotator.pyc
    /opt/opsware/bi.....
7625 /opt/opsware/bin/python /opt/opsware/agentproxy/lib/
    main.pyc.....
```

On Solaris, this component (specifically, the programs Hub, Adapter, and Agent Proxy) has seven running processes.

On Linux, execute the following command on the server running the Opsware Global File System Server.

```
# ps u -g $(cat /var/opt/opsware/hub/hub.pgrp)
```

Running this command should produce output similar to the following output:

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 8862 0.0 0.0 2436 1356 ? S Sep29 0:00 /opt/opsware/
    bin/python /opt/opsware/hub/bin/rotator.py /opt/opsware/
    j2sdk1.4.2_10/b.....
root 8868 0.1 1.8 1256536 76672 ? S Sep29 35:51 /opt/opsware/j2sdk1.4.2_
    10/bin/java -server -Xms64m -Xmx1024m -Dhub.kernel=Linux -
    Dh.....
root 8906 0.0 0.0 2412 1304 ? S Sep29 0:28 /opt/opsware/bin/python /opt/
    opsware/adapter/bin/adapter.....
root 8908 0.0 0.0 13088 684 ? S Sep29 0:10 /opt/opsware/adapter/Linux/
    bin/adapter.bin /var/opt/opsware/ogfs/mnt/ogfs -f -o
    none.....
root 8913 0.0 0.0 2308 1132 ? S Sep29 0:00 /opt/opsware/bin/python /opt/
    opsware/agentproxy/bin/rotator.pyc /opt/opsware/bin/
    pyt.....
```

```
root 8923 0.0 0.1 153120 6544 ? S Sep29 5:56 /opt/opsware/bin/python
/opt/opsware/agentproxy/lib/main.pyc.....
```

On Linux, this component (specifically, the programs Hub, Adapter, and Agent Proxy) has six running processes.

The Opware Global File System Server also supports a `status` option to the `init` script for both Linux and Solaris.

On Linux or Solaris, execute the following command on the server running the Opware Global File System Server to run this `status` option:

```
# /etc/opt/opsware/startup/hub status
```

Running this command should produce output similar to the following output:

```
Testing for presence of Hub process group file (/var/opt/opsware/hub/
hub.pgrp) ... OK
Testing that processes are running in Hub process group (8862) ... OK
Testing that OGFS is mounted ... OK
Testing that the OGFS authenticate file is present ... OK
OGFS is running
```

Opware Global File System Server Logs

The Hub logs are in the following files:

- `/var/log/opsware/hub/hub.log*`
- `/var/log/opsware/hub/hub.out*`

Conditions to monitor in the Hub logs:

- Strings containing ““Can't establish twist connection”

The Adapter logs are in the following files:

- `/var/log/opsware/adapter/adapter.err*`

The Agent Proxy logs are in the following files:

- `/var/log/opsware/agentproxy/agentproxy.err*`

Monitoring Processes for FUSE (Linux Only)

On Linux, execute the command on the server running the Opware Global File System Server:

```
# lsmod | grep -v grep | grep fuse
```

Running this command should produce output similar to the following output:


```
fuse          31196    2
```

FUSE logs messages in the following file:

- /var/log/messages

Monitoring Processes for the SunOS Kernel Module

On Solaris, the OGFS functionality relies on the SunOS kernel module.

Execute the command on the server running the Opsware Global File System Server:

```
# modinfo | grep -i opsware
```

Running this command should produce output similar to the following output:

```
137 1322cd8 43a9 272 1 ogdrv (Opsware GFS driver v1.13)
138 13ac227 338df 18 1 ogfs (Opsware Global Filesystem v1.14)
```

The Opsware Global File System Server logs messages related to SunOS kernel module in the following file:

- /var/adm/messages

Spoke Monitoring

The Spoke is the back-end component of the Opsware SAS Client. The Spoke, a Java RMI server, provides access to the files in the Opsware Global File System (OGFS) and provides access to run commands inside an OGFS session.

Spoke Ports

The Opsware Spoke uses port 8020.

Monitoring Processes for the Spoke

On Solaris, execute the command on the server running the Opsware Global File System Server:

```
# /usr/ucb/ps auxwww | grep -v grep | grep Spoke
```

Running this command should produce output similar to the following output:

```
root  4831  0.1  1.316426451168 pts/1  S   Jul 26  167:58  /opt/opsware/
j2sdk1.4.2_10/bin/java -server -Xms32m -Xmx256m
-Dbea.home=/opt/opsware/spoke/etc -Dspoke.home=/opt/
opsware/spoke -Dspoke.cryptodir=/var/opt/opsware/crypto/
spoke -Dspoke.logdir=/var/log/opsware/spoke
```

```
-Djava.util.logging.config.file=/opt/opsware/spoke/  
etc/logging.bootstrap  
-Dweblogic.security.SSL.ignoreHostnameVerification=true  
..... -classpath /opt/opsware/spoke/lib/HTTPClient-  
hacked.jar: ..... com.opsware.spoke.Spoke
```

On Linux, execute the command on the server running the Opware Global File System Server:

```
# ps -ef | grep -v grep | grep spoke
```

Running this command should produce output similar to the following output:

```
root 29191 1 0 Aug28 ? 01:12:11 /opt/opsware/j2sdk1.4.2_10/bin/  
java -server -Xms32m -Xmx256m -Dbea.home=/opt/opsware/  
spoke/etc -Dspoke.home=/opt/opsware/spoke  
-Dspoke.cryptodir=/var/opt/opsware/crypto/spoke  
-Dspoke.logdir=/var/log/opsware/spoke  
-Djava.util.logging.config.file=/opt/opsware/spoke/etc/  
logg
```

On Linux, the Spoke component has a single, running Java process.

Spoke Logs

The Spoke logs are in the following files:

- /var/log/opsware/spoke/spoke-*.log
- /var/log/opsware/spoke/stdout.log

Opware Gateway Monitoring

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.

For information about installing and configuring the Opware Gateway, see the “Opware Satellite Installation” chapter and the “Opware Gateway Properties File” appendix the *Opware® SAS Planning and Installation Guide*.

For information about maintaining the Opware Gateway in a satellite, see the *Opware® SAS Administration Guide*.

Opsware Gateway Ports

By default, the Opsware Gateway uses the following ports:

- 3002 (Core Gateway port – configurable)
- 3001 (Agent Gateway port – configurable)
- 3001 (Satellite Gateway port – configurable)

Monitoring Processes for the Opsware Gateway

In all configurations, the Opsware Gateway component has two running process – the Gateway process itself and its watchdog process.

On Solaris or Linux, execute the commands on the server running the Opsware Gateway component:

```
# ps -eaf | grep -v grep | grep opswgw | grep cgw
```

Running this command should produce output similar to the following output:

```
root 17092 1 0 Sep21 ? 00:00:00 [opswgw-watchdog-2.1.1: cgw0-C43]
--PropertiesFile /etc/opt/opsware/opswgw-cgw0-C43/
opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
root 17094 17092 0 Sep21 ? 02:23:21 [opswgw-gateway-2.1.1: cgw0-
C43] --PropertiesFile /etc/opt/opsware/opswgw-cgw0-C43/
opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
--Child true
```

```
# ps -eaf | grep -v grep | grep opswgw | grep agw
```

Running this command should produce output similar to the following output:

```
root 17207 1 0 Sep21 ? 00:00:00 [opswgw-watchdog-2.1.1: agw0-C43]
--PropertiesFile /etc/opt/opsware/opswgw-agw0-C43/
opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
root 17208 17207 0 Sep21 ? 01:18:54 [opswgw-gateway-2.1.1: agw0-
C43] --PropertiesFile /etc/opt/opsware/opswgw-agw0-C43/
opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
--Child true
```

In an Opsware Satellite facility on Solaris or Linux, execute the command on the server running the Opsware Satellite Gateway component:

```
# ps -eaf | grep -v grep | grep opswgw | grep <gateway-name>
```

Where <gateway-name> in this example is Sat1.

Running this command should produce output similar to the following output:

```
root 17092 1 0 Sep21 ? 00:00:00 [opswgw-watchdog-2.1.1: Sat1]
```

```
root 17094 17092 0 Sep21 ? 02:23:21 [opswgw-gateway-2.1.1: Sat1]
--PropertiesFile /etc/opt/opsware/opswgw-Sat1/
opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
--PropertiesFile /etc/opt/opsware/opswgw-Sat1/
opswgw.properties --BinPath /opt/opsware/opswgw/bin/opswgw
--Child true
```

Opware Gateway URL

https://occ.<data_center>/com.opsware.occ.gwadmin/index.jsp

(Log into the Opware SAS Web Client UI and select Gateway under Administration in the navigation panel.)

Opware Gateway Logs

The Opware Gateway logs are in the following files:

- /var/log/opsware/gateway-name/opswgw.log*

Conditions to monitor in the logs:

- Strings containing “ERROR”
- Strings containing “FATAL” (indicates that the process will end soon)

OS Build Manager Monitoring

The OS 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.

OS Build Manager Ports

The OS Build Manager uses the following ports:

- 1012 (HTTPS)
- 1017 (Opware SAS Build Agent)

Monitoring Processes for the OS Build Manager

In all configurations, the OS Build Manager component has a single running process.

On Solaris or Linux, execute the command on the server running the OS Build Manager component:

```
# ps -eaf | grep -v grep | grep buildmgr
```

Running this command should produce output similar to the following output:

```
root  2174  1  0  Sep27  ?  00:13:54  /opt/opsware/j2sdk1.4.2_10/bin/
      java -Xmx256m -Dbuildmgr -Djava.security.properties=/opt/
      opsware/buildmgr/etc/java.security -DDEBUG -DDEBUG_
      VERBOSE=1 -DLOG_OPTIONS=tTN -DLOG_FILE_THRESHOLD=10485760
      -DLOG_FILE_RETAIN_COUNT=7 -DLOG_
      CLASSES=com.opsware.buildmgr.OutputStreamLo
```

OS Build Manager URL

```
https://buildmgr.<data_center>:1012
```

The OS Build Manager UI is read-only and port 1012 for the UI is configurable.

OS Build Manager Logs

The OS Build Manager logs are in the following files:

- /var/log/opsware/buildmgr/buildmgr.log (Build Agent activities, OS provisioning activities)
- /var/log/opsware/buildmgr/*.request.log (Web Server log; one file per day; 90 logs maximum)
- /var/log/opsware/buildmgr/console.log
- /var/log/opsware/buildmgr/servers/<IP_address or machine_ID or MAC_address> (A per connection log)

Conditions to monitor in the logs: the string “Traceback”

OS Boot Server Monitoring

The OS 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.

These applications are installed by the Opware Installer but are not specific to Opware SAS. Monitor them by using standard system administration best practices for these applications.

OS Boot Server Ports

The OS Boot Server uses the following ports:

- 67 (UDP) (DHCP service)
- 69 (UDP) (TFTP service)

OS Boot Server Logs

The OS Boot Server does not generate its own logs. The OS 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. See also the `syslog.conf` file that was used to configure the Opware OS Boot Server to determine how the logging has been configured for this component.

OS Media Server Monitoring

The OS Media Server, part of the OS Provisioning feature, 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.

These applications are installed by the Opware Installer but are not specific to Opware SAS. Specifically, Opware Inc. provides a Samba package for Linux and Solaris that customers can use to install the OS Media Server. NFS services are provided by the operating system. Using the Opware Installer to install the OS Media Server configures NFS on Linux and Solaris.

Monitor the Samba SMB server and Sun Solaris NFS applications by using standard system administration best practices for these applications.

OS Media Server Ports

The OS Media Server uses the following ports:

- The portmapper used by NFS is port 111.
- Samba SMB uses ports 137, 138, 139, and 445.

OS Media Server Logs

The OS Media Server logs are in the following files:

- `/var/log/opsware/samba/log.smbd`
- `/var/log/opsware/samba/log.nmbd`

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

Chapter 7: Opware SAS Configuration

IN THIS CHAPTER

The topics covered in this section include:

- System Configuration
- Ways to Use Opware SAS Configuration Parameters

System Configuration

During the installation of an Opware core the Opware Installer sets specific system configuration parameters. In addition to the parameters that are set during installation, there are also many default values for the various system configuration parameters that should not be changed unless expressly directed to do so by Opware, Inc.

For information about how to use this function when you install an Opware core, see the *Opware[®] SAS Planning and Installation Guide*.



The Opware Agent reads the system configuration values at installation time only. If any of the configuration values change, the agent configuration must be updated manually. Contact Opware, Inc. Technical Support for help making these changes, or in making any other changes in the System Configuration area of Opware SAS.

Ways to Use Opware SAS Configuration Parameters

This section documents how to set specific parameters after you install an Opware core so that Opware SAS properly sends email alerts and displays the correct support contact information for your organization.

Where a value for a configuration parameter must be set for an installation of an Opware core, *Opware[®] SAS Planning and Installation Guide* provides instructions for setting the value. Set configuration values for those parameters as explicitly directed by the steps in the installation procedures.



Do not change other configuration values, unless explicitly directed to do so by this guide or by *Opware® SAS Planning and Installation Guide* or by your Opware, Inc. Support Representative.

After you install an Opware core, you should set several configuration parameter values that Opware SAS uses to send email notifications and alerts, and to display the Opware administrator contact information.

These values are set by selecting Administration ► System Configuration in the SAS Web Client.

Configuring Contact Information in the Opware Help

To configure the Opware administrator contact information that appears in Opware SAS Help page, perform the following steps:

1 In the Opware SAS core, log on as root to the server running the Opware Command Center Component (referred to as the Opware SAS Web Client in the UI).

2 Change directories to the following directory:

```
/etc/opt/opware/occ
```

3 Open the `psrvr.properties` file in a text editor.

4 In the `psrvr.properties`, change the values in the following fields to change the contact information in the SAS Web Client Help:

```
pref.occ.support.href
```

```
pref.occ.support.text
```

5 Save and exit from the file.

6 Restart the Opware Command Center component by entering the following command:

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

Configuring the Mail Server for a Facility

Perform the following steps in an Opware multimaster mesh to configure the mail server for the core running in each facility.

- 1** Log into the SAS Web Client as the admin user with the password you supplied during the interview. Log in by opening a browser and entering the IP address of the server running the SAS Web Client.

The SAS Web Client should be installed and listening. The SAS Web Client home page appears.

- 2** From the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 3** Under Select a Product, click the link for the facility name. The configuration page for the facility appears.

Opware components use the parameter `opware.mailserver` to determine the address of the mail server to use. If a value is not entered in the field, by default, the value of `opware.mailserver` is `smtp`. If managed servers are able to contact a mail server by using this name as the address, then you do not need to modify this parameter.

- 4** In the field, `opware.mailserver`, enter the host name of the mail server.
- 5** Click **Save** to apply the changes. The configuration page refreshes and a message appears that the update was successful.
- 6** From the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 7** Under Select a Product, click **Command Engine**.
- 8** In the field, `way.notification.email.fromAddr`, enter the From email address for the email messages that will be sent by the Command Engine to notify users about scheduled jobs.
- 9** Click **Save** to apply the changes.
- 10** Restart the Command Engine and SAS Web Client.
- 11** If Opware SAS is running in multimaster mode, restart the Model Repository Multimaster Component.

When restarting multiple Opware components, you must restart them in the correct order. See Chapter 5, “Starting an Opware SAS Core” on page 192 of this guide.

Setting Email Alert Addresses for an Opware Core



You should configure these email alert addresses before you install an Opware Agent on the servers in your operational environment because the Opware Agent on a managed server will only read this email configuration information the first time it contacts Opware SAS.

Perform the following steps to configure these email alert addresses. The Opware Installer installs an Opware core with placeholder values (EMAIL_ADDR) for these parameters.

- 1** Log into the SAS Web Client as the admin user with the password you supplied during the interview. Log in by opening a browser and entering the IP address of the server running the SAS Web Client.

The SAS Web Client should be installed and listening. The SAS Web Client home page appears.

- 2** From the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 3** Under Select a Product, click the Opware Agent link. The configuration page for the Opware Agent appears.
- 4** Configure the following required email alert addresses:
 - In the field, `acsbar.ErrorEmailAddr`, enter the address that Opware SAS will send warning emails to when any configuration tracking limit is exceeded (for example, when the configuration tracking feature stopped backing up configuration files and databases).
 - In the field, `acsbar.emailFromAddr`, enter the address that the Opware Agent will use as the email From address in the emails when Opware SAS detects a tracked configuration change.
Recommendation – use `agent@yourdomain.com`.
 - In the field, `CronbotAlertAddress`, enter the email address that the Opware Agent will use to alert the recipient about failed scheduled jobs.
 - In the field, `CronbotAlertFrom`, enter the email address that the Opware Agent will use as the email From address in the emails about failed scheduled jobs.
Recommendation – use `agent@yourdomain.com`.

- 5 Click **Save** to apply the changes. The configuration page refreshes and a message appears that the update was successful.

Configuring Email Alert Addresses for Multimaster

Perform the following steps to configure email alert addresses for multimaster. The Opsware Installer installs an Opsware core with placeholder values (EMAIL_ADDR) for these parameters.

- 1 Log into as the admin user with the password you supplied during the interview. Log in by opening a browser and entering the IP address of the server running the SAS Web Client.

The SAS Web Client should be installed and listening. The SAS Web Client home page appears.

- 2 From the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 3 Under Select a Product, click the Model Repository, Multimaster Component link. The configuration page for the Model Repository, Multimaster Component appears.
- 4 Configure the following email parameters:
 - In the field, sendMMErrorsTo, enter the email address to which multimaster conflicts will be sent.
 - In the field, sendMMErrorsFrom, enter the address that Opsware SAS will use as the email From address in the emails when multimaster conflicts are detected.
- 5 Click **Save** to apply the changes. The configuration page refreshes and a message appears that the update was successful.

Restart the Model Repository Multimaster Component in all Opsware cores in the multimaster mesh. See Chapter 5, “Starting an Opsware SAS Core Component” on page 192 of this guide.

Configuring Email Notification Addresses for CDR

You can set up email notification addresses for the Opsware Code Deployment & Rollback feature. When users request that a service operation or synchronization be performed on their behalf, an email notification is sent to the individuals assigned to perform the requested service operation or synchronization.

Perform the following steps to configure email notification addresses for CDR. The Opware Installer installs an Opware core with placeholder values (EMAIL_ADDR) for these parameters.

- 1** Log into the SAS Web Client as the admin user with the password you supplied during the interview. Log in by opening a browser and entering the IP address of the server running the SAS Web Client.

The SAS Web Client should be installed and listening. The SAS Web Client home page appears.

- 2** From the navigation panel, click System Configuration under Administration. The Select a Product page appears.
- 3** Click the link for the SAS Web Client. The configuration page appears, as Figure 7-1 shows.

Figure 7-1: CDR Email Notification Configuration Parameters

Modify configuration parameters for: Opware > Opware Command Center	
Name	Value
RackLocationMask: Show the Rack Location mask when managing datacenters	<input type="radio"/> Use default value: <i>no value</i> <input type="radio"/> Use value: <input type="text"/> ...
cds.requestfromaddress: E-mail for from address for a Code Deployment operation request	<input type="radio"/> Use default value: <i>no value</i> <input checked="" type="radio"/> Use value: <input type="text" value="support@xyz.com"/> ...
cds.requesttoaddress: Email address to which "request to perform an operation" are sent.	<input type="radio"/> Use default value: <i>no value</i> <input checked="" type="radio"/> Use value: <input type="text" value="support@xyz.com"/> ...
cds.supportaddress: E-mail for Code Deployment support	<input type="radio"/> Use default value: <i>no value</i> <input checked="" type="radio"/> Use value: <input type="text" value="support@xyz.com"/> ...
cds.supportorg: Code Deployment support organization name	<input type="radio"/> Use default value: <i>no value</i> <input checked="" type="radio"/> Use value: <input type="text" value="Opware Administrator"/> ...
cds.wayfrom: E-mail for from address for a Code Deployment Sequence report	<input type="radio"/> Use default value: <i>no value</i> <input checked="" type="radio"/> Use value: <input type="text" value="support@xyz.com"/> ...

- 4** Customize the following parameters to include the following email notification information:
 - In the field, cds.requesttoaddress, enter the email address to include in the To field of the email message for a request notification.

- In the field, `cds.requestfromaddress`, enter the email address to include in the From field of the email message for a request notification.
- In the field, `cds.wayfrom`, enter the email address to include in the From field of the email message sent following completion of a sequence.
- In the field, `cds.supportaddress`, enter the email address to include for a facilities' support organization or contact person.
- In the field, `cds.supportorg`, enter the display name of a facilities' support organization.

5 Click **Save** to apply the changes. The configuration page refreshes and a message appears that the update was successful.

6 Restart the Command Engine and the Model Repository Multimaster Component.

When you restart multiple Opsware SAS components, you must restart them in the correct order.

Appendix A: Permissions Reference

IN THIS APPENDIX

This section discusses the following topics:

- Permissions Required for the SAS Web Client
- Permissions Required for the Opsware SAS Client
- Script Execution Permissions
- Predefined User Group Permissions
- Code Deployment User Groups

Permissions Required for the SAS Web Client

The following table lists the feature permissions according to tasks that can be performed with the SAS Web Client.

Table A-1: Permissions Required for SAS Web Client Tasks

TASK	FEATURE PERMISSION
OS PROVISIONING	
Prepare OS	Wizard: Prepare OS
Edit OS nodes	Operating Systems
View servers in the server pool	Server Pool
CONFIGURATION TRACKING	
Create or edit tracking policy	Configuration Tracking Managed Servers and Groups
Reconcile tracking policy	Configuration Tracking Managed Servers and Groups
Perform configuration backup	Configuration Tracking Managed Servers and Groups

Table A-1: Permissions Required for SAS Web Client Tasks (continued)

TASK	FEATURE PERMISSION
View backup history, restore queue	Configuration Tracking Managed Servers and Groups
Enable or disable tracking	Configuration Tracking Managed Servers and Groups
SERVER MANAGEMENT	
Edit server properties	Managed Servers and Groups
Edit server network properties	Managed Servers and Groups
Edit server custom attributes	Managed Servers and Groups
Deactivate server	Deactivate
Delete server	Managed Servers and Groups
Clone server	Managed Servers and Groups
Re-assign customer	Managed Servers and Groups
View servers (read-only access)	Managed Servers and Groups
Run server communications test	Managed Servers and Groups
Lock servers	Managed Servers and Groups
Set scheduled job to refresh server list	Allow Run Refresh Jobs
REPORTS	
Create or view reports	Data Center Intelligence Reports
MANAGE ENVIRONMENT	
Create or edit customer	Customers
Create or edit facility	Facilities
IP RANGES AND RANGE GROUPS	
IP Ranges	IP Ranges and Range Groups Model: Hardware Model: Opware
IP Range Groups	IP Ranges and Range Groups Model: Hardware Model: Opware

Table A-1: Permissions Required for SAS Web Client Tasks (continued)

TASK	FEATURE PERMISSION
SYSTEM CONFIGURATION	
Manage users and groups	(Administrators group only)
Define server attributes	Server Attributes
Run system diagnosis tools	System Diagnosis
Manage Opware System configuration	Configure Opware
Run Opware multimaster tools	Multimaster
Gateway management	Manage Gateway
OTHER TASKS	
Run custom extension	Wizard: Custom Extension
Run scripts	See "Script Execution Permissions" on page 301.
Deploy code	See "Code Deployment User Groups" on page 308.

Permissions Required for the Opware SAS Client

The tables in this section summarize the permissions required for the Opware SAS Client features.

More Information for Security Administrators

In some organizations, security administrators work with many applications and do not specialize in Opware SAS. To learn about Opware SAS quickly, security administrators can refer to the following documentation:

- Glossary in the *Opware[®] SAS User's Guide: Server Automation* - The Glossary defines terms that are unique to Opware SAS, such as Snapshot and Audit Template.
- "Process Overview for Security Administration" on page 79 - This short section lists the overall tasks for setting up security in Opware SAS.

Application Configuration Management Permissions

Table A-2 specifies the Application Configuration Management permissions required by users to perform specific actions in the Opware SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?



In addition to the feature permissions listed in Table A-2, every user action also requires the Managed Servers and Groups feature permission.

In Table A-2, the Server Permission column is for the servers referenced by the Application Configuration or Application Configuration Template. Server permissions are specified by the Customer, Facility, and Device Groups permissions in the SAS Web Client. In Table A-2, the Customer Permission column is for the customers associated with Application Configurations or Application Configuration Templates.

To perform an action, the user requires several permissions. For example, to attach an application configuration to a server, the user must have the following permissions:

- Manage Application Configurations: Read
- Manage Configuration Templates: Read
- Manage Installed Configuration and Backups on Servers: Read & Write
- Managed Servers and Groups
- Read & Write permissions to the facility, device group, and customer of the server
- Read permission for the customer of the Application Configuration

Table A-2: Application Configuration Management Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Application Configuration			

Table A-2: Application Configuration Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Create Application Configuration	Manage Application Configurations: Read & Write and Manage Configuration Templates: Read	None	Read & Write
View Application Configuration	Manage Application Configurations: Read & Write and Manage Configuration Templates: Read	None	Read
Edit Application Configuration	Manage Application Configurations: Read & Write and Manage Configuration Templates: Read	None	Read & Write
Delete Application Configuration	Manage Application Configurations: Read & Write and Manage Configuration Templates: Read	None	Read & Write

Table A-2: Application Configuration Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Specify Template Order	Manage Application Configurations: Read & Write and Manage Configuration Templates: Read	None	Read & Write
Attach Application Configuration to Server	Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read & Write	Read & Write	Read
Set Application Configuration Values on Server	Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read & Write	Read & Write	Read

Table A-2: Application Configuration Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Push Application Configuration to Server	Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read & Write	Read & Write	Read
Schedule Application Configuration Push	Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read & Write	Read & Write	Read
Scan Configuration Compliance	Allow Configuration Compliance Scan: Yes and Manage Application Configurations: Read and Manage Configuration Templates: Read	Read	Read

Table A-2: Application Configuration Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Schedule Application Configuration Audit	Allow Configuration Compliance Scan: Yes and Manage Application Configurations: Read and Manage Configuration Templates: Read	Read	Read
Roll Back (Revert) Application Configuration Push	Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read & Write	Read & Write	Read
Application Configuration Templates			
Create Application Configuration Template	Manage Configuration Templates: Read & Write	None	Read & Write
View Application Configuration Template	Manage Configuration Templates: Read & Write	None	Read
Edit Application Configuration Template	Manage Configuration Templates: Read & Write	None	Read & Write
Delete Application Configuration Template	Manage Configuration Templates: Read & Write	None	Read & Write

Table A-2: Application Configuration Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Load (Import) Application Configuration Template	Manage Application Configurations: Read & Write and Manage Configuration Templates: Read & Write	None	Read & Write
Set Application Configuration Template to Run as Script	Manage Configuration Templates: Read & Write	None	Read & Write
Compare Two Application Configuration Templates	Manage Configuration Templates: Read	None	Read
Compare Application Configuration Template Against Actual Configuration File (Preview)	Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read	Read	Read

Table A-3 lists the actions that users can perform for each Application Configuration Management permission. Table A-3 has the same data as Table A-2, but is sorted by feature permission. Although not indicated in Table A-3, the Managed Servers and Groups permission is required for all OS provisioning actions.

For security administrators, Table A-3 answers this question: If a user is granted a particular feature permission, what actions can the user perform?

Table A-3: User Actions Allowed by Application Configuration Management Permissions

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Allow Configuration Compliance Scan: Yes and Manage Application Configurations: Read and Manage Configuration Templates: Read	Scan Configuration Compliance	Read	Read
	Schedule Application Configuration Audit	Read	Read
Manage Application Configurations: Read & Write and Manage Configuration Templates: Read	Create Application Configuration	None	Read & Write
	Delete Application Configuration	None	Read & Write
	Edit Application Configuration	None	Read & Write
	Specify Template Order	None	Read & Write
	View Application Configuration	None	Read
Manage Application Configurations: Read & Write and Manage Configuration Templates: Read & Write	Load (Import) Application Configuration Template	None	Read & Write

Table A-3: User Actions Allowed by Application Configuration Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read	Compare Application Configuration Template Against Actual Configuration File (Preview)	Read	Read
Manage Application Configurations: Read and Manage Configuration Templates: Read and Manage Installed Configuration and Backups on Servers: Read & Write	Attach Application Configuration to Server	Read & Write	Read
	Push Application Configuration to Server	Read & Write	Read
	Roll Back (Revert) Application Configuration Push	Read & Write	Read
	Schedule Application Configuration Push	Read & Write	Read
	Set Application Configuration Values on Server	Read & Write	Read
Manage Configuration Templates: Read	Compare Two Application Configuration Templates	None	Read
Manage Configuration Templates: Read & Write	Create Application Configuration Template	None	Read & Write
	Delete Application Configuration Template	None	Read & Write
	Edit Application Configuration Template	None	Read & Write

Table A-3: User Actions Allowed by Application Configuration Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	CUSTOMER PERMISSION (APP CONFIG, APP CONFIG TEMPLATE)
Manage Configuration Templates: Read & Write (cont.)	Set Application Configuration Template to Run as Script	None	Read & Write
	View Application Configuration Template	None	Read

Device Group Permissions

To use the Device Groups feature in the Opware SAS Client, you must have the permissions described in the Table A-4. For a list of tasks that require the Model Public Device Group permission, see Table A-11.

Table A-4: Device Groups Feature Permissions

USER ACTION	FEATURE PERMISSION
Creating a public static device group	Manage Public Device Group
Creating a public dynamic device group	Manage Public Device Group
Adding a server to a public static device group	Manage Public Device Group
Adding a server to a public dynamic device Group	Manage Public Device Group
Removing a server from a public static device group	Manage Public Device Group
Removing servers from a public dynamic device group	Manage Public Device Group
Moving a public device group	Manage Public Device Group
Duplicating a public device group	Manage Public Device Group
Deleting a public device group	Manage Public Device Group

ODAD Permissions

To use Opsware Discovery and Deployment (ODAD) in the Opsware SAS Client, you must have the permissions described in the Table A-5.

Table A-5: ODAD Feature Permissions

USER ACTION	FEATURE PERMISSION
Deploy (Install) Agent with ODAD	Allow Deploy Agent: Yes
Scan Network with ODAD	Allow Scan Network: Yes
View Servers Running Agents	Managed Servers and Groups

In addition to the feature permissions listed in the preceding table, you must have Read permissions on the following:

- Customer named Opsware
- Facility that has the servers targeted for Agent deployment.

To use ODAD on Windows managed servers, you also need Read permissions on the following:

- Customer that owns the server running the Windows Agent Deployment Helper (ADH)
- Facility that has the server running the Windows ADH

Job Permissions

To manage jobs in the Opsware SAS Client, you must have the permissions described in the Table A-6. When you select the Edit All Jobs permission, the View All Jobs permission is automatically selected.

Table A-6: Job Management Permissions

USER ACTION	FEATURE PERMISSION
Enable Approval Integration	Manage Approval Integration
Set Job Types Requiring Approval	Manage Approval Integration

Table A-6: Job Management Permissions

USER ACTION	FEATURE PERMISSION
Invoke JobService API Methods to Manage Blocked (Pending Approval) Jobs (This action is performed by customized software on the backend, not by end-users logged onto the SAS Client.)	Edit All Jobs View All Jobs
End (Cancel) Job	Edit All Jobs View All Jobs
Delete Schedule	Edit All Jobs View All Jobs

Patch Management for Windows Permissions

Table A-7 specifies the Patch Management permissions required by users to perform specific actions in the Opware SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?



In addition to the feature permissions listed in Table A-7, every user action also requires the Managed Servers and Groups feature permission.

In Table A-7, most of the entries in the User Action column correspond to menu items in the Opware SAS Client. In addition to feature permissions, server permissions are required on the managed servers affected by the patching operation.



If the Allow Install Patch permission is set to Yes, then the Manage Patch and the Manage Patch Policies permissions are automatically set to Read.

Table A-7: Windows Patch Management Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Patches		
Install Patch (Available)	Allow Install Patch: Yes Manage Patch: Read	Read & Write
Uninstall Patch (Available)	Allow Uninstall Patch: Yes and Manage Patch: Read	Read & Write
Install Patch (Limited Availability)	Allow Install Patch: Yes Manage Patch: Read & Write	Read & Write
Uninstall Patch (Limited Availability)	Allow Uninstall Patch: Yes and Manage Patch: Read & Write	Read & Write
Open Patch (View Patch)	Manage Patch: Read	N/A
Change Patch Properties	Manage Patch: Read & Write	N/A
Import Patch	Manage Patch: Read & Write and Package	N/A
Import Patch Database	Manage Patch: Read & Write	N/A
Export Patch	Manage Patch: Read and Package	N/A
Export Patch	or Allow Install Patch: Yes and Package: Yes	N/A
Export Patch	or Allow Uninstall Patch: Yes and Package	N/A
Export Patch	or Manage Policy: Read and Package	N/A
Delete Patch	Manage Patch: Read & Write	N/A

Table A-7: Windows Patch Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Patch Policies and Exceptions		
Remediate Policy	Allow Install Patch: Yes	Read & Write
Open Patch Policy (View)	Manage Patch Policy: Read	N/A
Add Patch to Patch Policy	Manage Patch: Read and Manage Patch Policy: Read & Write	N/A
Remove Patch from Patch Policy	Manage Patch Policy: Read & Write	N/A
Set Exception	Allow Install Patch: Yes	Read & Write
Set Exception	or Allow Uninstall Patch: Yes	Read & Write
Copy Exception	Allow Install Patch: Yes	Read & Write
Copy Exception	or Allow Uninstall Patch: Yes	Read & Write
Attach Patch Policy to Server (or Device Group)	Manage Patch Policy: Read	Read & Write
Detach Patch Policy from Server (or Device Group)	Manage Patch Policy: Read	Read & Write
Create Patch Policy	Manage Patch Policy: Read & Write	N/A
Delete Patch Policy	Manage Patch Policy: Read & Write	N/A
Change Patch Policy Properties	Manage Patch Policy: Read & Write	N/A
Patch Compliance Rules		
Edit Patch Products (Patch Configuration window)	Manage Patch Compliance Rules: Yes	N/A
Scan Patch Compliance	Manage Patch Policy: Read	N/A
Schedule a Patch Policy Scan	Manage Patch Compliance Rules: Yes	N/A
Change Default Patch Availability	Manage Patch Compliance Rules: Yes	N/A

Table A-7: Windows Patch Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Change Patch Policy Compliance Rules	Manage Patch Compliance Rules: Yes	N/A
View Patch Policy Compliance Rules	Manage Patch Policy: Yes	N/A

Table A-8 lists the actions that users can perform for each Patch Management permission. Table A-8 has the same data as Table A-7, but is sorted by feature permission. Although it is not indicated in Table A-8, the Managed Servers and Groups permission is required for all Patch Management actions.

For security administrators, Table A-8 answers this question: If a user is granted a particular feature permission, what actions can the user perform?

Table A-8: User Actions Allowed by Windows Patch Management Permissions

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Allow Install Patch: Yes	Copy Exception	Read & Write
	Remediate Policy	Read & Write
	Set Exception	Read & Write
Allow Install Patch: Yes and Manage Patch: Read	Install Patch (Available)	Read & Write
	Uninstall Patch (Available)	Read & Write
Allow Install Patch: Yes and Manage Patch: Read & Write	Install Patch (Limited Availability)	Read & Write
	Uninstall Patch (Limited Availability)	Read & Write

Table A-8: User Actions Allowed by Windows Patch Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Allow Install Patch: Yes and Package: Yes	Export Patch	N/A
Allow Uninstall Patch: Yes	Copy Exception	Read & Write
	Set Exception	Read & Write
Allow Uninstall Patch: Yes and Package	Export Patch	N/A
Allow Uninstall Patch: Yes and Manage Patch: Read	Uninstall Patch	Read & Write
Manage Patch Compliance Rules: Yes	Change Default Patch Availability	N/A
	Change Patch Policy Compliance Rules	N/A
	Edit Patch Products (Patch Configuration window)	N/A
	Schedule a Patch Policy Scan	N/A
Manage Patch Policy: Read	Attach Patch Policy to Server (or Device Group)	Read & Write
	Detach Patch Policy from Server (or Device Group)	Read & Write
	Open Patch Policy (View)	N/A
Manage Patch Policy: Read & Write	Change Patch Policy Properties	N/A
	Create Patch Policy	N/A
	Delete Patch Policy	N/A
	Remove Patch from Patch Policy	N/A
Manage Patch Policy: Yes	View Patch Policy Compliance Rules	N/A

Table A-8: User Actions Allowed by Windows Patch Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Manage Patch: Read	Open Patch (View Patch)	N/A
	Scan Patch Compliance	
Manage Patch: Read & Write	Change Patch Properties	N/A
	Delete Patch	N/A
	Import Patch Database	N/A
Manage Patch: Read & Write and Package	Import Patch	N/A
Manage Patch: Read and Manage Patch Policy: Read & Write	Add Patch to Patch Policy	N/A
Manage Patch: Read and Package	Export Patch	N/A
Manage Policy: Read and Package	Export Patch	N/A

Patch Management for Unix Permissions

Table A-9 specifies the Patch Management permissions required by users to perform specific actions in the Opsware SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?



In addition to the feature permissions listed in Table A-9, every user action also requires the Managed Servers and Groups feature permission.

In Table A-9, most of the entries in the User Action column correspond to menu items in the Opsware SAS Client. In addition to feature permissions, server permissions are required on the managed servers affected by the patching operation.



If the Allow Install Patch permission is set to Yes, then the Manage Patch and the Manage Patch Policies permissions are automatically set to Read.

Table A-9: Unix Patch Management Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Patches		
Install Patch (Available)	Allow Install Patch: Yes Manage Patch: Read	Read & Write
Uninstall Patch (Available)	Allow Uninstall Patch: Yes and Manage Patch: Read	Read & Write
Install Patch (Limited Availability)	Allow Install Patch: Yes Manage Patch: Read & Write	Read & Write
Uninstall Patch (Limited Availability)	Allow Uninstall Patch: Yes and Manage Patch: Read & Write	Read & Write
Open Patch (View Patch)	Manage Patch: Read	N/A
Change Patch Properties	Manage Patch: Read & Write	N/A
Export Patch	Manage Patch: Read and Package	N/A
Export Patch	or Allow Install Patch: Yes and Package: Yes	N/A
Export Patch	or Allow Uninstall Patch: Yes and Package	N/A
Export Patch	or Manage Policy: Read and Package	N/A
Delete Patch	Manage Patch: Read & Write	N/A

Table A-10 lists the actions that users can perform for each Patch Management permission. Table A-10 has the same data as Table A-9, but is sorted by feature permission. Although it is not indicated in Table A-10, the Managed Servers and Groups permission is required for all Patch Management actions.

For security administrators, Table A-10 answers this question: If a user is granted a particular feature permission, what actions can the user perform?

Table A-10: User Actions Allowed by Unix Patch Management Permissions

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Allow Install Patch: Yes	Copy Exception	Read & Write
	Remediate Policy	Read & Write
	Set Exception	Read & Write
Allow Install Patch: Yes and Manage Patch: Read	Install Patch (Available)	Read & Write
	Uninstall Patch (Available)	Read & Write
Allow Install Patch: Yes and Manage Patch: Read & Write	Install Patch (Limited Availability)	Read & Write
	Uninstall Patch (Limited Availability)	Read & Write
Allow Install Patch: Yes and Package: Yes	Export Patch	N/A
Allow Uninstall Patch: Yes	Copy Exception	Read & Write
	Set Exception	Read & Write
Allow Uninstall Patch: Yes and Package	Export Patch	N/A
Manage Patch: Read	Open Patch (View Patch)	N/A
Manage Patch: Read & Write	Change Patch Properties	N/A
	Delete Patch	N/A
	Import Patch Database	N/A

Table A-10: User Actions Allowed by Unix Patch Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Manage Patch: Read & Write and Package	Import Patch	N/A
Manage Patch: Read and Manage Policy: Read & Write	Add Patch to Policy	N/A
Manage Patch: Read and Package	Export Patch	N/A
Manage Policy: Read and Package	Export Patch	N/A

Software Management Permissions

Table A-11 specifies the Software Management permissions required by users to perform specific actions in the SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?

If a customer is assigned to a folder, then customer constraints might limit the objects that can be associated with a software policy contained in the folder. For a list of tasks affected by these constraints, see “Customer Constraints, Folders, and Software Policies” on page 75.

Table A-11: Software Management Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Software Policy			
Create Software Policy	Manage Software Policy: Read & Write	N/A	Write

Table A-11: Software Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Delete Software Policy	Manage Software Policy: Read & Write	N/A	Write
Open Software Policy (View)	Manage Software Policy: Read	N/A	Read
Edit Software Policy Properties	Manage Software Policy: Read & Write	N/A	Write
Add Packages	Manage Software Policy: Read & Write Manage Packages: Read	N/A	Folder containing the software policy: Write
Add RPM Packages	Manage Software Policy: Read & Write Manage Packages: Read	N/A	Folder containing the software policy: Write
Add Patches	Manage Software Policy: Read & Write Manage Patches: Read	N/A	Folder containing the software policy: Write
Add Application Configurations	Manage Software Policy: Read & Write Manage Application Configuration: Read	N/A	Folder containing the software policy: Write
Add Software Policies	Manage Software Policy: Read & Write	N/A	Folder containing the software policy: Write
Remove Packages	Manage Software Policy: Read & Write	N/A	Write

Table A-11: Software Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Remove RPM Packages	Manage Software Policy: Read & Write	N/A	Write
Remove Patches	Manage Software Policy: Read & Write	N/A	Write
Remove Application Configurations	Manage Software Policy: Read & Write	N/A	Write
Remove Software Policies	Manage Software Policy: Read & Write	N/A	Write
Attach Software Policy	Manage Software Policy: Read Allow Attach/Detach Software Policy: Yes Model Public Device Groups: Yes (This permission is required if you are attaching the software policy to a public device group)	Read & Write	Read
Detach Software Policy	Manage Software Policy: Read Allow Attach/Detach Software Policy: Yes Model Public Device Groups: Yes (This permission is required if you are attaching the software policy to a public device group)	Read & Write	Read

Table A-11: Software Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Remediate	Manage Software Policy: Read Allow Remediate Servers: Yes Model Public Device Groups: Yes (Required if you remediate a public device group)	Read & Write	Read
Install Software	Manage Software Policy: Read Allow Attach/Detach Software Policy: Yes Allow Remediate Servers: Yes Model Public Device Groups: Yes (Required if you remediate a public device group)	Read & Write	Read
Install Software Template	Manage Software Policy: Read Allow Install Software Policy Templates: Yes Model Public Device Groups: Yes (Required if you install a software policy template on a public device group)	Read & Write	Read
Run ISM Control	Manage Software Policy: Read Allow Run ISM Control: Yes Model Public Device Groups: Yes (Required if you run ISM Control on a public device group)	Read & Write	Read

Table A-11: Software Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Duplicate Zip Package	Manage Software Policy: Read & Write	N/A	Write
Edit ZIP Installation Directory	Manage Software Policy: Read & Write	N/A	Write
Scan Software Compliance	N/A	Read	N/A
Rename Software Policy	Manage Software Policy: Read & Write	N/A	Write
Cut Software Policy	Manage Software Policy: Read & Write	N/A	Write
Copy Software Policy	Manage Software Policy: Read	N/A	Read
Paste Software Policy	Manage Software Policy: Read & Write	N/A	Source Folder: Read (for copy and paste) Source Folder: Write (for cut and paste) Destination Folder: Write
Move Software Policy	Manage Software Policy: Read & Write	N/A	Source Folder: Write Destination Folder: Write
Folder			
Create Folder	N/A	N/A	Write
Delete Folder	N/A	N/A	Write
Open Folder	N/A	N/A	Read

Table A-11: Software Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
View Folder Properties	N/A	N/A	Read
Edit Folder Properties	N/A	N/A	Write
Manage Folder Permissions	N/A	N/A	Edit Folder Permissions
Cut Folder	N/A	N/A	Write
Copy Folder	N/A	N/A	Read
Paste Folder	N/A	N/A	Source Folder: Read (for copy and paste) Source Folder: Write (for cut and paste) Destination Folder: Write
Move Folder	N/A	N/A	Source Folder: Write Destination Folder: Write
Rename Folder	N/A	N/A	Write
Package			
Import Package	Manage Package: Read & Write	N/A	Write
Export Package	Manage Package: Read	N/A	Read
Open Package (View)	Manage Package: Read	N/A	Read
Edit Package Properties	Manage Package: Read & Write	N/A	Read
Delete Package	Manage Package: Read & Write	N/A	Write
Rename Package	Manage Package: Read & Write	N/A	Write

Table A-11: Software Management Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Cut Package	Manage Package: Read & Write	N/A	Write
Paste Package	Manage Package: Read & Write	N/A	Source Folder: Read (for copy and paste) Source Folder: Write (for cut and paste) Destination Folder: Write
Move Package	Manage Package: Read & Write	N/A	Source Folder: Write Destination Folder: Write

Table A-12 lists the actions that users can perform for each Software Management permission. Table A-12 has the same data as Table A-11, but is sorted by feature permission. For security administrators, Table A-12 answers this question: If a user is granted a particular feature permission, what actions can the user perform?

Table A-12: User Actions Allowed by Software Management Permissions

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Manage Software Policy: Read & Write	Create Software Policy	N/A	Write
	Delete Software Policy	N/A	Write
	Edit Software Policy	N/A	Write
	Rename Software Policy	N/A	Write
	Cut Software Policy	N/A	Write
	Paste Software Policy	N/A	Write
	Move Software Policy	N/A	Write
	Remove Packages	N/A	Write
	Remove Patches	N/A	Write
	Remove Application Configurations	N/A	Write
	Remove Software Policy	N/A	Write
	Duplicate ZIP packages	N/A	Write
Manage Software Policy: Read	Open Software Policy (View)	N/A	Read
	Copy Software Policy Properties	N/A	Read
Manage Software Policy: Read & Write And Manage Package: Read	Add Packages Add RPM Packages	N/A	Folder containing the software policy: Write Folder containing the package: Read

Table A-12: User Actions Allowed by Software Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Manage Software Policy: Read & Write And Manage Patches: Read	Add Patches	N/A	Folder containing the software policy: Write Folder containing the patch: Read
Manage Software Policy: Read & Write And Manage Application Configuration: Read	Add Application Configurations	N/A	Folder containing the software policy: Write Folder containing the application configuration: Read
Manage Software Policy: Read & Write	Add Software Policies	N/A	Folder containing the software policy: Write Folder containing the software policy to be added to another software policy: Read
Manage Software Policy: Read & Write	Remove Packages	N/A	Write
	Remove RPM Packages		
	Remove Patches	N/A	Write
	Remove Application Configurations	N/A	Write
	Remove Software Policies	N/A	Write

Table A-12: User Actions Allowed by Software Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Manage Software Policy: Read And Allow Attach/Detach Software Policy: Yes And Model Public Device Groups: Yes (Required if you are attaching the software policy to a public device group)	Attach Software Policy	Read & Write	Read
	Detach Software Policy	Read & Write	Read
Manage Software Policy: Read And Allow Remediate Servers: Yes And Model Public Device Groups: Yes (Required if you remediate a public device group)	Remediate	Read & Write	Read

Table A-12: User Actions Allowed by Software Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Manage Software Policy: Read And Allow Attach/Detach Software Policy: Yes And Allow Remediate Servers: Yes And Model Public Device Groups: Yes (Required if you remediate a public device group)	Install Software	Read & Write	Read
Manage Software Policy: Read And Allow Install Software Policy Templates: Yes And Model Public Device Groups: Yes (Required if you install a software policy template on a public device group)	Install Software Template	Read & Write	Read

Table A-12: User Actions Allowed by Software Management Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSIONS
Manage Software Policy: Read And Allow Run ISM Control: Yes And Model Public Device Groups: Yes (Required if you run ISM Control on a public device group)	Run ISM Control	Read & Write	Read
Manage Package: Read & Write	Import Package	N/A	Write
	Delete Package	N/A	Write
	Rename Package	N/A	Write
	Cut Package	N/A	Write
	Paste Package	N/A	Write
	Move Package	N/A	Write
Manage Package: Read & Write	Edit Package Properties	N/A	Read
Manage Package: Read	Export Package	N/A	Read
	Open Package (View)	N/A	Read

Audit and Remediation Permissions

Table A-13 specifies the Audit and Remediation permissions required by users to perform specific actions in the SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?



In addition to the feature permissions listed in Table A-13, every user action also requires the Managed Servers and Groups feature permission.

Server Permissions for Audit and Remediation

Audit and Remediation actions require both feature and server feature permissions. For example, the Create Audit action requires the feature permission “Manage Audit: Read & Write” and the Managed Servers and Groups feature permission. This action also needs Read permission on the server referenced by the Audit. In Table A-13, the Server Permission column is for the servers referenced by the Audit or Snapshot Specification – depending on the action. Server permissions are specified by the customer, facility, and device groups permissions in the SAS Web Client.

If an Audit and Remediation object (such as a Snapshot Specification) references multiple servers, at least Read permission is required for all servers referenced. Otherwise, the object cannot be viewed or modified.

Audit and Remediation objects are not directly associated with customers and facilities, but customer and facility permissions do control access to servers which are referenced by Audit and Remediation objects, such as Snapshot Specifications and Audits.

Global Shell Permissions for Audit and Remediation

For the actions that access a managed server’s file system, the Global Shell `readServerFilesystem` permission is required. For example, the `readServerFilesystem` permission is required to create a Snapshot Specification with rules that include the files of a managed server. Such Rules include and Application Configurations, Custom Scripts, COM+ objects, File System, IIS Metabase entries, and Windows Registry.

Other types of selection criteria require the corresponding Global Shell permissions:

- `readServerRegistry`
- `readServerComplus`
- `readServerMetabase`

To grant these permissions, run the `aaa` command within a Global Shell session.

Audit and Remediation User Action Permissions

The following table lists typical Audit and Remediation user actions and the permissions required to perform them.

Table A-13: Audit and Remediation Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Snapshot Specification			
View contents of Snapshot Specification	Manage Snapshot Specification: Read	N/A	Read
Schedule and run a Snapshot Specification	Manage Snapshot Specification: Read	N/A	Read
Create, edit, and delete Snapshot Specification	Manage Snapshot Specification: Read & Write	N/A	Read & Write
Create Application Configuration Rule	Manage Snapshot Specification: Read & Write	writeServer Filesystem	Read & Write
Create COM+ Rule	Manage Snapshot Specification: Read & Write	readServerComplus	Read & Write
Create Custom Script Rule	Manage Snapshot Specification: Read & Write Allow Create Custom Script Policy Rules: Yes.	writeServer Filesystem	Read & Write
Create File System Rule	Manage Snapshot Specification: Read & Write	writeServer Filesystem	Read & Write

Table A-13: Audit and Remediation Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Create IIS Metabase Rule	Manage Snapshot Specification: Read & Write	readServerMetabase	Read & Write
Create Registry Rule	Manage Snapshot Specification: Read & Write	readServerRegistry	Read & Write
Link Audit Policy into Snapshot Specification	Manage Snapshot Specification: Read & Write	N/A	Read & Write
Import Audit Policy into Snapshot Specification	Manage Snapshot Specification: Read & Write Allow Create Task Specific Policy: Yes	N/A	Read & Write
Save As Audit Policy	Manage Snapshot Specification: Read & Write	N/A	Read & Write
Snapshots			
View, list contents of a Snapshot	Manage Snapshot: Read	N/A	Read
Delete Snapshot results	Manage Snapshot: Read & Write	N/A	Read & Write
Detach Snapshot from a server	Allow General Snapshot Management	N/A	Read
Remediate Snapshot results	Manage Snapshot: Yes Allow Remediate Audit/ Snapshot Results: Yes	N/A	Read & Write

Table A-13: Audit and Remediation Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Remediate Snapshot Results: Application Configuration	Manage Snapshot: Yes Allow Remediate Audit/Snapshot Results: Yes	writeServerFilesystem	Read & Write
Remediate Snapshot Results: COM+	Manage Snapshot: Yes Allow Remediate Audit/Snapshot Results: Yes	readServerComplus	Read & Write
Remediate Snapshot Results: Custom Scripts	Manage Snapshot: Yes Allow Remediate Audit/Snapshot Results: Yes	writeServerFilesystem	Read & Write
Remediate Snapshot Results: File System	Manage Snapshot: Yes Allow Remediate Audit/Snapshot Results: Yes	writeServerFilesystem	Read & Write
Remediate Snapshot Results: Metabase	Manage Snapshot: Yes Allow Remediate Audit/Snapshot Results: Yes	readServerMetabase	Read & Write
Remediate Snapshot Results: Registry	Manage Snapshot: Yes Allow Remediate Audit/Snapshot Results: Yes	readServerRegistry	Read & Write
Audits			
View an Audit	Manage Audit: Read	N/A	Read
Schedule and run an Audit	Manage Audit: Read	N/A	Read
Create, edit, and delete an Audit	Manage Audit: Read & Write	N/A	Read & Write
Create Application Configuration Rule	Manage Audit: Read & Write	writeServerFilesystem	Read & Write

Table A-13: Audit and Remediation Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Create COM+ Rule	Manage Audit: Read & Write	readServerComplus	Read & Write
Create Custom Script Rule	Manage Audit: Read & Write Allow Create Custom Script Policy Rules: Yes	writeServerFilesystem	Read & Write
Create File System Rule	Manage Audit: Read & Write	writeServerFilesystem	Read & Write
Create IIS Metabase Rule	Manage Audit: Read & Write	readServerMetabase	Read & Write
Create Registry Rule	Manage Audit: Read & Write	readServerRegistry	Read & Write
Link Audit Policy into an Audit	Manage Audit: Read & Write	N/A	Read & Write
Import Audit Policy into an Audit	Manage Audit: Read & Write Allow Create Task Specific Policy: Yes	N/A	Read & Write
Save as Audit Policy	Manage Audit: Read & Write	N/A	Read & Write
Audit Results			
View Audit results	Manage Audit Results: Read	N/A	Read
Delete Audit results	Manage Audit Results: Read & Write	N/A	Read & Write
Remediate Audit Results	Manage Audit Results: Read & Write Allow Remediate Audit/ Snapshot Results: Yes	N/A	Read & Write

Table A-13: Audit and Remediation Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Remediate Audit Results: Application Configuration	Manage Audit Results: Read & Write Allow Remediate Audit/Snapshot Results: Yes	writeServerFilesystem	Read & Write
Remediate Audit Results: Custom Script Rule	Manage Audit Results: Read & Write Allow Remediate Audit/Snapshot Results: Yes	writeServerFilesystem	Read & Write
Remediate Audit Results: COM+	Manage Audit Results: Read & Write Allow Remediate Audit/Snapshot Results: Yes	readServerComplus	Read & Write
Remediate Audit Results: File System	Manage Audit Results: Read & Write Allow Remediate Audit/Snapshot Results: Yes	writeServerFilesystem	Read & Write
Remediate Audit Results: IIS Metabase	Manage Audit Results: Read & Write Allow Remediate Audit/Snapshot Results: Yes	readServerMetabase	Read & Write
Remediate Audit Results: Windows Registry	Manage Audit Results: Read & Write Allow Remediate Audit/Snapshot Results: Yes	readServerRegistry	Read & Write

Table A-14 lists the actions that users can perform for each Audit and Remediation permission. Table A-14 has the same data as Table A-13, but is sorted by feature permission. Although it is not indicated in Table A-14, the Managed Servers and Groups permission is required for all Audit and Remediation actions.

For security administrators, Table A-14 answers this question: If a user is granted a particular feature Audit and Remediation permission, what actions can the user perform?

Table A-14: User Actions Allowed by Audit and Remediation Permissions

FEATURE PERMISSION	USER ACTION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Allow Create Custom Script Rule Policy: No and Manage Audit: Read	View Custom Script Rule: Audit	N/A	Read
Allow Create Custom Script Rule Policy: Yes and Manage Audit: Read & Write	Create Custom Script Rule: Audit	writeServer Filesystem	Read & Write
Allow Create Custom Script Rule Policy: No and Manage Snapshot: Read & Write	View Custom Script Rule: Snapshot	N/A	Read
Allow Create Custom Script Rule Policy: Yes and Manage Snapshot: Read & Write	Create Custom Script Rule: Snapshot	writeServer Filesystem	Read & Write

Table A-14: User Actions Allowed by Audit and Remediation Permissions (continued)

FEATURE PERMISSION	USER ACTION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Allow Create Task Specific Policy: No and Manage Audit: Read and Write	Link Audit Policy Into Audit	N/A	Read & Write
Allow Create Task Specific Policy: Yes and Manage Audit: Read & write	Import Audit Policy Into Audit	N/A	Read & Write
Allow General Snapshot Management: Yes	Detach Snapshot from a server	N/A	Read
Allow Remediate Audit/Snapshot Results: No and Manage Audit or Manage Snapshot Specification: Read	View Audit or Snapshot Specification Results, No Remediation	N/A	Read
Allow Remediate Audit/Snapshot Results: Yes and Manage Audit or Manage Snapshot Specification: Read & Write	Remediate Audit/Snapshot Results	N/A	Read & Write

Table A-14: User Actions Allowed by Audit and Remediation Permissions (continued)

FEATURE PERMISSION	USER ACTION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Allow Remediate Audit/Snapshot Results: Yes and Manage Audit or Manage Snapshot Specification: Read & Write	Remediate Application Configuration Rule	writeServerFilesystem	Read & Write
	Remediate COM+ Rule	readServerComplus	Read & Write
	Remediate Custom Script Rule Registry Rule	writeServerFilesystem	Read & Write
	Remediate File System Rule	readServerMetabase	Read & Write
	Remediate IIS Metabase Rule	readServerRegistry	Read & Write
	Remediate Windows Registry Rule	writeServerFilesystem	Read & Write
Manage Audit: Read	View, schedule, run Audit	N/A	Read
Manage Audit: Read & Write	Create, edit, delete Audit	N/A	Read & Write
	Save Audit as Audit Policy	N/A	Read & Write
	Link Audit Policy into Audit	N/A	Read & Write
	Create Application Configuration Rule	writeServerFilesystem	Read & Write
	Create COM+ Rule	readServerComplus	Read & Write
	Create File System Rule	writeServerFilesystem	Read & Write
	Create IIS Metabase Rule	readServerMetabase	Read & Write
	Create Window Registry Rule	readServerRegistry	Read & Write

Table A-14: User Actions Allowed by Audit and Remediation Permissions (continued)

FEATURE PERMISSION	USER ACTION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Manage Audit: Read & Write and Allow Create Custom Script Policy Rules: Yes	Create Custom Scripts Rule	writeServerFilesystem	Read & Write
Manage Audit Results: Read	View Audit Results	N/A	Read
Manage Audit Results: Read & Write	Delete Audit Results	N/A	Read & Write
Manage Snapshot Specification: Read	View, schedule, run Snapshot Specification	N/A	Read
Manage Snapshot Specification: Read & Write	Create, edit, and delete Snapshot Specification	N/A	
	Save Snapshot Specification as Audit Policy	N/A	
	Link Audit Policy Into Audit	N/A	Read & Write
	Create Application Configuration Rule	writeServerFilesystem	Read & Write
	Create COM+ Rule	readServerComplus	Read & Write
	Create File System Rule	writeServerFilesystem	Read & Write
	Create IIS Metabase Rule	readServerMetabase	Read & Write
	Create Windows Registry Rule	readServerRegistry	Read & Write

Table A-14: User Actions Allowed by Audit and Remediation Permissions (continued)

FEATURE PERMISSION	USER ACTION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Manage Snapshot Specification: Read & Write and Create Custom Script Policy Rule	Create Custom Rule for Snapshot Specification	writeServerFilesystem	Read & Write
Manage Snapshot: Read	View contents of Snapshot	N/A	Read
Manage Snapshot: Read & Write	Delete Snapshot results	N/A	Read & Write
Manage Audit Policy: Read	View contents of Audits and Snapshot Specifications	N/A	Read
Manage Audit Policy: Read & Write	Create, edit, and delete Audit Policy.	N/A	Read & Write
	Create Application Configuration Rule	writeServerFilesystem	Read & Write
	Create COM+ Rule	readServerComplus	Read & Write
	Create File System Rule	writeServerFilesystem	Read & Write
	Create IIS Metabase Rule	readServerMetabase	Read & Write
	Create Windows Registry Rule	readServerRegistry	Read & Write

Table A-14: User Actions Allowed by Audit and Remediation Permissions (continued)

FEATURE PERMISSION	USER ACTION	GLOBAL SHELL PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
Manage Audit Policy: Read & Write and Allow Create Custom Script Policy Rule	Create Custom Script Rule	writeServer Filesystem	Read & Write

Visual Application Manager Permissions

Table A-15 specifies the Visual Application Manager (VAM) permissions required to perform specific actions in the SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?

In Table A-15, most of the entries in the User Action column correspond to menu items in the SAS Client. In addition to feature permissions, server read permissions are required on the managed servers affected by the analyze operation, such as permissions to open a Remote Terminal or a Remote Desktop Client, open the Server Explorer, and open a Global Shell session from the Visual Application Manager.



VAM permissions required to scan a server are the same for both physical servers and virtual servers.

For details on this feature, see the “Visual Application Manager” chapter in the *Opware® SAS User’s Guide: Application Automation*.

Table A-15: Visual Application Manager Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	SOURCE SERVER PERMISSION (CUSTOMER, FACILITY)
Launch the Visual Application Manager	Allow Analyze: Yes Read & Write	Read
Generate a Scan – regular or virtual servers	Allow Analyze: Yes Read & Write	Read

Virtualization Director Permissions

Table A-16 specifies the Virtualization Director permission required to perform specific actions in the SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?

In Table A-16, most of the entries in the User Action column correspond to menu items in the SAS Client.

In addition to feature permissions, server read permissions are required on hypervisor servers. Once you have created a new virtual server, then its permissions are treated just like a regular physical server, including OS Provisioning.

For details on this feature, see the Virtualization Director chapter in the *Opware® SAS User's Guide: Server Automation*.

Table A-16: Virtualization Director Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	HYPERVISOR SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
View the Virtual Servers feature in the SAS Client navigation panel View virtualization object in a virtual server's device explorer	View Virtual Servers: Yes (Note: If this permission is set to No, you will still be able to see virtual servers in the All Managed Servers list.)	Read on the hypervisor server
Refresh a hypervisor server	View Virtual Servers: Yes	Read on the hypervisor server
Create, modify, or remove a virtual server (Solaris local zone)	View Virtual Servers: Yes Manage Virtual Servers: Yes	Read on the hypervisor server
Start and stop a Solaris local zone	View Virtual Servers: Yes Manage Virtual Servers: Yes Administer Virtual Servers: Yes	Read on the hypervisor server

Table A-17 lists the actions that users can perform for each Virtualization Director permission. Table A-17 has the same data as Table A-16, but is sorted by feature permission. For security administrators, Table A-17 answers this question: If a user is granted a particular feature Virtualization Director permission, what actions can the user perform?

Table A-17: User Actions Allowed by Virtualization Director Permissions

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)
View Virtual Servers: Yes Manage Virtual Servers: No Administer Virtual Server: No	View the Virtual Servers feature in the SAS Client navigation panel and virtual servers in the Contents pane. View virtualization object in a virtual server's device explorer Refresh a hypervisor server	Read on the hypervisor server
View Virtual Servers: Yes Manage Virtual Servers: Yes Administer Virtual Server: No	Create, modify, or remove a virtual server (Solaris local zone or VMware ESX VM)	Read on the hypervisor server
View Virtual Servers: Yes Manage Virtual Servers: Yes Administer Virtual Servers: Yes	Start and stop a Solaris local zone	Read on the hypervisor server

Visual Packager Permissions

Table A-18 specifies the Visual Packager permissions required by users to perform specific actions in the Opware SAS Client. For security administrators, the table answers this question: To perform a particular action, what permissions does a user need?

In addition to the feature permissions, users also need access to servers. In Table A-18, the Source Server Permission column is for the servers referenced when the package is created or modified. The server in the Packaging Server column is where the IDK is installed. These server permissions are specified by the Customer and Facility permissions in the SAS Web Client.



In addition to the feature permissions listed in Table A-18, every user action also requires Write permission on the folder.

For details on this feature, see the *Opsware[®] SAS Policy Setter's Guide*.

Table A-18: Visual Packager Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	SOURCE SERVER PERMISSION (CUSTOMER, FACILITY)	PACKAGING SERVER PERMISSION (CUSTOMER, FACILITY)
Create a Package from a Server	Allow Create Package: Yes Manage Software Policy: Read & Write Manage Services: Read & Write	Read	Read
Create a Package from a Snapshot	Allow Create Package: Yes Manage Snapshot on Server: Read Manage Software Policy: Read & Write	Read	Read
Create a Package from an Audit Result	Allow Create Package: Yes Manage Audit: Read Manage Software Policy: Read & Write	Read	Read

OS Provisioning Permissions

The following section describes the OS Provisioning permissions required by users to perform specific actions in the Opware SAS. For security administrators, the following table answers this question: To perform a particular action, what permissions does a user need?

In Table A-19, the Server Permission column is for the servers referenced by the OS sequence or installation profile. Server permissions are specified by the Customer, Facility, and Device Groups permissions in the Opware SAS Web Client.

With the OS Provisioning feature in the Opware SAS Web Client, in order to create and save an OS sequence you must save it in a folder, so you will need write permissions to the folder.

See “Customer Permissions and Folders” on page 73 in this chapter for more information.

Table A-19: OS Provisioning Permissions Required for User Actions

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSION
OS Sequence			
Create OS Sequence	Manage OS Sequence: Read & Write	None	Write
View OS Sequence	Manage OS Sequence: Read	None	Read
Edit OS Sequence	Manage OS Sequence: Read & Write	None	Write
Delete OS Sequence	Manage OS Sequence: Read & Write	None	Write
Run OS Sequence (From server or from OS sequences)	Manage OS Sequence: Read and Allow Execute OS Sequence: Yes	Read & Write	Read

Table A-19: OS Provisioning Permissions Required for User Actions (continued)

USER ACTION	FEATURE PERMISSION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER PERMISSION
View unprovisioned servers	SAS Web Client permission: Server Pool	Read	N/A
OS Installation Profile			
Create, edit, delete OS installation profile	Wizard: Prepare OS	Read	N/A
Unprovisioned Server List			
View servers in the unprovisioned server list	Server Pool	N/A	N/A

Table A-20 lists the actions that users can perform for each OS Provisioning permission. Table A-20 has the same data as Table A-19, but is sorted by feature permission.

For security administrators, Table A-20 answers this question: If a user is granted a particular feature permission, what actions can the user perform?

Table A-20: User Actions Allowed in the SAS Client by OS Provisioning Permissions

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER
Manage OS Sequence: Read	View OS sequence	Read	Read
Manage OS Sequence: Read & Write	Run OS sequence	Write	Write
	Create OS sequence	Read	Write
Manage OS Sequence: Read & Write			

Table A-20: User Actions Allowed in the SAS Client by OS Provisioning Permissions (continued)

FEATURE PERMISSION	USER ACTION	SERVER PERMISSION (CUSTOMER, FACILITY, DEVICE GROUP)	FOLDER
Allow Execute OS Sequence: Yes	Run OS sequence	Write	Read
Allow Execute OS Sequence: No	View OS sequence	N/A	Read
Manage OS Sequence: Read Allow execute OS Sequence: Yes	Run OS sequence	Write	Read
Manage OS Sequence: Read Allow Execute OS Sequence: No	View OS sequence	Read	Read
Manage OS Sequence: Write Allow Execute OS Sequence: Yes	Run OS sequence Edit OS sequence	Write	Write
Manage OS Sequence: Write Allow Execute OS Sequence: No	Edit OS sequence	Read	Write
Wizard: Prepare OS	Create, edit, delete OS installation profile	Read	N/A
Server Pool	View servers in the unprovisioned server list	Read	N/A

Script Execution Permissions

The following table provides an overview of the most common script management and script execution tasks, and displays the permissions required to perform a task. Because each task is performed on a specific type of script (My Scripts, Shared Script, or Ad-Hoc Script), the table also lists the permissions according to the type of script.

Table A-21: Permissions Required for Script Tasks

SCRIPT TASK	SCRIPT TYPE	REQUIRED PERMISSIONS FOR SCRIPT TASK AND TYPE	COMMENTS
SCRIPT MANAGEMENT TASKS			
View list of available scripts.	My Script, Shared Script	Scripts	
Create or upload and store a script.	My Script	Scripts	
Edit, delete, or view a stored script.	My Script	Scripts	
View version history of a stored script.	My Script	Scripts	
Create or upload and store a script.	Shared Script	Scripts Edit Shared Scripts	
Edit, delete, or view a stored script.	Shared Script	Scripts Edit Shared Scripts	
View version history of a stored script.	Shared Script	Scripts Edit Shared Scripts	

Table A-21: Permissions Required for Script Tasks (continued)

SCRIPT TASK	SCRIPT TYPE	REQUIRED PERMISSIONS FOR SCRIPT TASK AND TYPE	COMMENTS
SCRIPT EXECUTION TASKS			
Execute a script (as root or local system).	Shared Script	Wizard: Run Scripts Read & Write (for servers associated with customers, facilities, or device groups)	A Shared Script always executes on a server as root or local system.
Execute a script (requires a password).	My Script	Wizard: Run Scripts Scripts Read & Write (for servers associated with customers, facilities, or device groups)	With Wizard: Run Scripts and Scripts permissions, execution of a My Script requires the use of a password.
Execute a script (as root or local system).	My Script	Wizard: Run Scripts Scripts Run My Script As Root Read & Write (for servers associated with customers, facilities, or device groups)	With Run My Script As Root permission, no password is required. Without this permission, the user can still execute the script, but only with a password.

Table A-21: Permissions Required for Script Tasks (continued)

SCRIPT TASK	SCRIPT TYPE	REQUIRED PERMISSIONS FOR SCRIPT TASK AND TYPE	COMMENTS
Create (or upload) and then execute an Ad-Hoc Script (requires a password).	Ad-Hoc Script	Wizard: Run Scripts Scripts Read & Write (for servers associated with customers, facilities, or device groups)	With Wizard: Run Scripts and Scripts permissions, a password is required to execute an Ad-Hoc Script.
Create (or upload) and then execute an Ad-Hoc Script (as root/local system).	Ad-Hoc Script	Wizard: Run Scripts Scripts Run My Script As Root Read & Write (for servers associated with customers, facilities, or device groups)	With Wizard: Run Scripts, Scripts, and Run My Script As Root permissions, an Ad-Hoc Script executes on the servers as root/local system (without a password).
View execution results data.	My Script, Shared Script, Ad-Hoc Script	• Wizard: Run Scripts	The Wizard: Run Scripts permission allows a user to view results information for any executed script.

Predefined User Group Permissions

The following table lists the permissions of the predefined user groups for the features in the SAS Web Client. An X in a table cell indicates that the group has permission to use the feature. The headings in the table columns abbreviate the names of the user groups as follows:

- **Basic:** Basic Users

- **Inter:** Intermediate Users
- **Adv:** Advanced Users
- **OSA:** Opware System Administrators
- **Admin:** Administrators

Table A-22: SAS Web Client Permissions of the Predefined User Groups

FEATURE NAME	BASIC	INTER	ADV	OSA	ADMIN
FEATURE TAB					
Configuration Tracking	X	X	X		
Configure Opware				X	
Customers	X	X	X		X
DNS	X	X	X		
Data Center Intelligence Reports			X	X	
Facilities	X	X	X		X
IP Ranges and Range Groups	X	X	X		
ISM Controls	X	X	X		
Manage Gateway				X	
Managed Servers and Groups	X	X	X	X	
Model: Hardware	X	X	X		
Model: Opware			X		
Model: Service Levels	X	X	X		
Multimaster				X	
Operating Systems		X	X		
Scripts	X	X	X	X	
Server Attributes			X	X	
Server Pool		X	X		
System Diagnosis			X	X	
Wizard: Custom Extension			X	X	
Wizard: Prepare OS	X	X	X		

Table A-22: SAS Web Client Permissions of the Predefined User Groups (continued)

FEATURE NAME	BASIC	INTER	ADV	OSA	ADMIN
Wizard: Run Scripts	X	X	X	X	
OTHER TAB					
Edit Shared Scripts			X	X	
Run My Scripts as Root	X	X	X	X	
Deactivate		X	X		
Allow Run Refresh Jobs					
Manage Public Device Groups				X	
Model Public Device Groups					
View All Jobs					
Edit All Jobs					

Only the Administrator group also has permission to manage Opware users and user groups, a feature not listed on the SAS Web Client tabs.

The following table lists the permissions of the predefined user groups for the Opware SAS Client features.

The table cells contain the following abbreviations:

- **R:** Read (only)
- **RW:** Read & Write
- **Y:** Yes
- **N:** No or None

Table A-23: Opware SAS Client Permissions of the Predefined User Groups

FEATURE NAME	BASIC	INTER	ADV	OSA	ADMIN
APPLICATION CONFIGURATION					
Configuration	N	R	RW	N	N
Configuration Files	N	R	RW	N	N
Configuration on Servers	N	R	RW	N	N

Table A-23: Opsware SAS Client Permissions of the Predefined User Groups (continued)

FEATURE NAME	BASIC	INTER	ADV	OSA	ADMIN
Allow Check Consistency on Servers	N	N	Y	N	N
COMPLIANCE					
Audit Templates	N	R	RW	N	N
Audit Results	N	R	RW	N	N
Snapshot Templates	N	R	RW	N	N
Snapshots (specific to servers)	N	R	RW	N	N
Selection Criteria	N	R	RW	N	N
Allow General Snapshot Management	N	Y	Y	N	N
VISUAL PACKAGER					
Allow Create Package	N	N	Y	N	N
AGENT DEPLOYMENT					
Allow Deploy Agent	N	N	Y	N	N
Allow Scan Network	N	N	Y	N	N
PATCH MANAGEMENT					
Manage Patch	N	N	RW	N	N
Manage Patch Policy	N	N	RW	N	N
Allow Install Patch	N	N	Y	N	N
Allow Uninstall Patch	N	N	Y	N	N
Manage Patch Compliance Rules	N	N	N	N	N

When Opsware SAS is first installed, default permissions are assigned to the top-level folders of the SAS Web Client. The following table lists these default permissions. The table uses the following abbreviations for permissions:

- **L**: List Contents of Folder
- **R**: Read Objects Within Folder
- **W**: Write Objects Within Folder
- **P**: Edit Folder Permissions

Table A-24: Default Top-Level Folder Permissions of the Predefined User Groups

FOLDER	BASIC	INTER	ADV	OSA	ADMIN
/	L	L	W	L	P
/Opsware		L	L	L	P
/Opsware/Tools		L	L	L	P
Opsware/Tools/Agent Deployment Helper			R	W	P
/Opsware/Tools/ISMTOOL		R	W		P
/Package Repository		R	W		P
/Package Repository/All AIX		R	W		P
/Package Repository/All AIX/AIX <version>		R	W		P
/Package Repository/All HP-UX		R	W		P
/Package Repository/All HP-UX/HP-UX <version>		R	W		P
/Package Repository/All Red Hat Linux		R	W		P
Package Repository/All Red Hat Linux/Red Hat Linux <version>		R	W		P
/Package Repository/All SunOS		R	W		P
/Package Repository/All SunOS/SunOS <version>		R	W		P
/Package Repository/All SuSE Linux		R	W		P

Table A-24: Default Top-Level Folder Permissions of the Predefined User Groups (continued)

FOLDER	BASIC	INTER	ADV	OSA	ADMIN
/Package Repository/All SuSE Linux/ SuSE Linux <version>		R	W		P
/Package Repository/All Windows		R	W		P
/Package Repository/All Windows/ Windows <version>		R	W		P

Code Deployment User Groups

The following tables describe the capabilities of the Code Deployment user groups. For more information, see the Accessing Code Deployment & Rollback section of the *Opware® SAS User's Guide: Server Automation*.

Table A-25: Special Code Deployment User Groups

CODE DEPLOYMENT USER GROUP	DESCRIPTION
Super User	Can define, request, or perform any code deployment operation on hosts designated for either staging or production. Because a Super User can perform operations on hosts associated with any customer, only a few users should belong to this group.
History Viewer	Can view a log of operations (service operations, synchronizations and sequences) that have been previously executed from the Code Deployment feature. Viewing this information can help you determine the status of particular deployment operations, and whether they completed successfully.

Table A-26: Service User Groups

CODE DEPLOYMENT USER GROUP	DESCRIPTION
Service Editor	Can define a service, and modify or delete service definitions.

Table A-26: Service User Groups (continued)

CODE DEPLOYMENT USER GROUP	DESCRIPTION
Production Service Performer	Can directly perform or request performance of service operations on hosts designated for use in production.
Staging Service Performer	Can directly perform or request performance of service operations on hosts designated for use in staging.
Production Service Requester	Can request performance of service operations on hosts designated for use in production.
Staging Service Requester	Can request performance of service operations on hosts designated for use in staging.

Table A-27: Synchronization User Groups

CODE DEPLOYMENT USER GROUP	DESCRIPTION
Synchronization Editor	Can define a synchronization, and modify or delete the synchronization definition.
Synchronization Performer	Can directly perform or request performance of a synchronization action.
Synchronization Requester	Can request performance of a synchronization action.

Table A-28: Sequence User Groups

CODE DEPLOYMENT USER GROUP	DESCRIPTION
Sequence Editor	Can define a sequence, and modify or delete the sequence definition.
Production Sequence Performer	Can directly perform or request performance of a sequence of actions on hosts designated for use in production.
Staging Sequence Performer	Can directly perform or request performance of a sequence of actions on hosts designated for use in staging.

Table A-28: Sequence User Groups (continued)

CODE DEPLOYMENT USER GROUP	DESCRIPTION
Production Sequence Requester	Can request performance of a sequence of actions on hosts designated for use in production.
Staging Sequence Requester	Can request performance of a sequence of actions on hosts designated for use in staging.

Appendix B: Software Repository Replicator Setup

IN THIS APPENDIX

After you install an Opsware core in multimaster mode, you can set up replication for the Software Repository in a facility.

This section discusses the following topics:

- Overview of the Software Repository Replicator
- Prerequisites for Using the Software Repository Replicator
- Software Repository Replicator Configuration

Overview of the 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 allows you to have redundant copies of Software Repositories and thereby helps to ensure that all packages remain available even when a Software Repository goes offline.

Prerequisites for Using the Software Repository Replicator

Before you set up the Software Repository Replicator, you must meet the following prerequisites:

- SSH must be installed on the source and target Software Repositories.
- Port 22 must be open on the firewalls.

- Passwordless SSH as root must be enabled between the source and target repositories.

Software Repository Replicator Configuration

By default, the Opware Installer installs the software you need to set up Software Repository replication when you install the multimaster Software Repository.

From the source core, use the `replicator.conf` file found in the `/etc/opt/opware/replicator` directory to configure the Software Repository Replicator.



To set up Software Repository replication, you do *not* need to modify the `replicator.conf` files in the target cores. However, you must specify to replicate the directory `/etc/opt/opware/replicator/` to all the target cores. You specify which target cores to replicate to by entering them in the host chain section of the `replicator.conf` file. When you specify replication to these target cores, the `replicator.conf` file will propagate to the Software Repositories in the target cores. See the bullet about defining host chains on page 313. See “Sample Software Repository Replicator Configuration” on page 313 in this chapter for information about how to specify target cores in a host chain.

In this file, you must specify the following settings:

- The values for `User`, `Timestampdir`, and `SSH_PATH`.
The Software Repository Replicator keeps timestamps of when it runs.
- For each directory that you want to replicate, specify the `Directory` or `WordDirectory` tag.
The Software Repository Replicator parses the `Directory` tags, and these are ignored by the Software Repository. Therefore, you can use the `Directory` tag to replicate files that are not served by the Software Repository. `WordDirectory` tags are parsed by both the Package Replicator and the Software Repository.

You should specify these tags for each directory you want to replicate because the Software Repository Replicator is not the only process that parses the `replicator.conf` file. Some of the other processes that parse the

`replicator.conf` file only use `WordDirectory` as a backup repository and ignore entries labeled "Directory."

To specify the entire contents of the Software Repository for replication, enter `<word_root>/<facility_name>` as `WordDirectory`, for example, `/var/opt/opsware/word/C28`. Do not specify `/var/opt/opsware/word/mmword_local` because it is a symlink to `/var/opt/opsware/word/<facility_name>`.

Do *not* replicate the directories for `mmword_cache` and `mmword_local`.

The Software Repository should ignore directories that are not actual software repositories while serving files. For example, files should not be served from the directory, `/etc/opt/opsware/mm_wordbot`.

- Specify the replication rate in seconds.

Define host chains. Make sure that the host names you specify are the actual host names (that is, the same that the `hostname` command returns).

For example, `hostA hostB hostC` means that a directory will be replicated from `hostA` to `hostB` to `hostC`.

For example `hostA hostB, hostA hostC` means that a directory will be replicated from `hostA` to `hostB` and from `hostA` to `hostC`.

In these examples, if you want the packages on each Software Repository host backed up, all the hosts have to be in the same multimaster mesh.

Verify that you can use passwordless SSH to connect from the source host to the destination host as it is specified for each host chain in the `replicator.conf` file (that is, if you specify FQDN, try to connect with SSH with FQDN even if `host.subdomain` resolves to the correct location.)

After you configure the Software Repository Replicator, you must re-start the replicator so that it will automatically re-read its configuration file. At each destination core, wait for the time period you specified in the `replicator.conf` file before you re-start the replicator.

To re-start the replicator, enter the following command on the server running the Software Repository component:

```
/etc/init.d/opsware-sas [start/stop] replicator
```

Sample Software Repository Replicator Configuration

User: root

Timestampdir: `/var/opt/opsware/replicator`

```
SSH_PATH: /usr/bin/ssh
Directory: 60 /etc/opt/opsware/mm_word
Chain: theword01.subdomain1.domain.com
      theword01.subdomain2.domain.com
Chain: theword01.subdomain1.domain.com
      theword01.subdomain3.domain.com
WordDirectory: 60 /var/opt/opsware/word/facility1
Chain: theword01.subdomain1.domain.com
      theword01.subdomain2.domain.com
WordDirectory: 60 /var/opt/opsware/word/facility2
Chain: theword01.subdomain2.domain.com
      theword01.subdomain3.domain.com
WordDirectory: 60 /var/opt/opsware/word/facility3
Chain: theword01.subdomain3.domain.com
      theword01.subdomain1.domain.com
```

Index

A

aaa utility	88
accessing, realm information	139
adding	
users to a user group	82
users to Administrator group	89
users to CDR user groups	102
Administrator group	
adding, users	89
removing, users	90
security administrator	79
agent-server architecture	27
Application Configuration Management, overview	63
auditverify tool	186
authentication, external	94

B

backup, deleting files	195
Boot Server	
defined	27, 30
logs	179
Build Agent, defined	27, 33
Build Manager	
defined	27, 30
logs	179

C

CDR. <i>See</i> Code Deployment & Rollback.	
code deployment	
configuring, email alert addresses	245
user groups	308
code deployment user groups	
adding users	102
Command Engine	
defined	26, 27
logs	179
scripts	30
system diagnostic tests	163
configuration	
Opsware SAS configuration parameters	241

Software Repository Replicator	312
configuration files	
modifying	95
configuring	
contact information	242
email alert addresses for multimaster	245
email alert addresses for Opsware core	244
email notification addresses for CDR	245
JAAS login module	100
mail server	243
password policy parameters	90
conflicts	
alert emails	128
causes	110
error messages	129
overview	109
prevention	112
resolving	118, 124
constraints	
customer and folder	75
contacting, Opsware support	158
content management, tools	51
conventions used in the guide	16
creating	
user groups	82
users	80
creating, Manual updates	152

D

Data Access Engine	
defined	27, 30
logs	179
multiple	198
reassigning	199
<i>See also</i> Multimaster Central Data Access Engine.	
system diagnostic tests	161
Data Center Intelligence reporting	
required permissions	250
deleting	
users	81
deleting, backup files	195

diagnosing, problems159
 digital186
 dormant, Opware Agents32

E

editing
 user information81
 email alert addresses
 CDR245
 multimaster245
 Opware core244
 enabling, realm information137

F

facilities
 defined135
 definition of26
 multiple105
 primary109
 viewing, information137

G

Global Shell
 setting, permissions87

I

importing, external LDAP users101
 importing, server certificate from external LDAP ...99
 Inbound, Model Repository Multimaster Component .
 31
 installations
 multiple Data Access Engine198
 types26
 installing
 patch43
 integrating, Opware SAS with AIX and HP-UX ...57
 IP range groups
 required permissions250
 IP ranges
 required permissions250

J

JBoss180

L

LDAP directory
 importing, external users101
 importing, server certificate99
 process for using external LDAP95
 supported external directory servers94
 using, external authentication94
 log
 digital signatures186
 logs
 about178
 Boot Server179
 Build Manager179
 Command Engine179
 configuring180, 188
 Data Access Engine179
 Global Shell Audit182
 JBoss180
 managed servers
 Global Shell logs182
 Media Server179
 Model Repository180
 Model Repository Multimaster Component ...180
 Opware Agents180
 SAS Web Client180
 Software Repository180
 Software Repository Multimaster Component .181
 Software Repository Replicator181
 Web Services Data Access Engine181

M

manage environment, required permissions250
 Manual updates
 creating152
 defined146
 overview150
 Software Repository Cache, applying to154
 uploading, Microsoft utilities155
 Media Server
 defined27, 30
 logs179
 Model Repository
 defined25, 27, 31
 logs180
 Model Repository Multimaster Component
 defined27, 31
 Inbound31
 logs180
 Outbound31
 system diagnostic tests163

- model-based control 25
 - modifying
 - Web Services Data Access Engine configuration file 95
 - monitoring remote server access 185
 - multimaster
 - alert emails during conflicts 128
 - central 109
 - configuring, email alert addresses 245
 - configuring, mail server 243
 - conflicts 109
 - designating the Central Data Access Engine .. 200
 - error messages in multimaster conflicts 129
 - installation 26
 - mesh 108, 113
 - mode 108
 - network administration 128
 - preventing conflicts 112
 - tools 113
 - multimaster central 109
 - Multimaster Central Data Access Engine 200
 - multimaster, tools 113, 117
- N**
- network administration 128
- O**
- On-demand updates
 - defined 146
 - overview 150
 - Opware Agent
 - defined 26, 27
 - dormant 32
 - Installer 32
 - logs 180
 - overview 31
 - Opware Agent Installer 32
 - Opware Command Center
 - defined 27
 - overview 33
 - Opware components
 - internal and external names 194
 - overview 29
 - running, system diagnosis 164
 - Opware Discovery and Agent Deployment
 - permissions required 260, 261
 - Opware Discovery and Agent Deployment, overview 61
 - Opware Gateway
 - defined 28, 34
 - Opware Global File System
 - defined 35
 - Opware guides
 - contents 15
 - conventions used 16
 - documentation set 17
 - icons in guide, explained 17
 - Opware SAS
 - agent-server architecture 27
 - components 29
 - components overview 27
 - configuration 241
 - configuration parameters 241
 - configuring, contact information 242
 - configuring, email alert addresses 244
 - core technology 24
 - documentation set 17
 - environment 24
 - integrating with AIX and HP-UX 57
 - model-based control 25
 - multimaster mode 108
 - multiple facilities 105
 - overview 21
 - related documentation 17
 - security 36
 - software provisioning 48
 - system diagnosis 159
 - tools 51
 - troubleshooting 158
 - types of users 23
 - Opware SAS Client
 - overview 32
 - Opware SAS Web Client
 - overview 32
 - Opware Satellite
 - accessing, realm information 139
 - definition 26
 - linked to cores 26
 - manual update 146
 - on-demand updates 146
 - overview 133
 - permissions, required 136
 - Software Repository Cache, overview 145
 - OS Build Agent. *See* Build Agent.
 - OS provisioning
 - required permissions 249
 - Outbound, Model Repository Multimaster Component 31

P

password policy parameters, configuring	90
patch management	
installing, patches	43
uninstalling, patches	45
updating, Microsoft patch	47
uploading, patches	41
permissions	
customer and server	73
delegate	89
delegated	74
folder	74, 79
folder and customer	83
IP ranges and IP range groups, required for	250
manage environment, required for	250
ODAD, required for	260, 261
OS Provisioning, required for	249
other tasks, required for	251
reports, required for	250
SAS Client permissions for user groups	303
SAS Web Client permissions for user groups	303
script execution, required for	301
script management and execution, required for	301
server management, required for	250
setting	
customer permissions	83
facility permissions	83
feature permissions	85
Global Shell permissions	87
other feature permissions	87
SAS Client feature permissions	86
server group permissions	84
system configuration, required for	251
viewing, user's permissions	81
Virtualization Director, required for	293
virtualization director, required for	294
prerequisites	
Software Repository Replicator	311
preventing, conflicts	112
preview reconcile	49
Primary Data Access Engine	199
primary facility	109
Python	30

R

realms	
defined	135
enabling realm information	137
viewing realm information	139

reassigning, Data Access Engine	199
remediate	50
removing	
users from Administrator group	90
resolving	
conflicts by object	118
conflicts by transaction	124
running, system diagnosis	164

S

SAS Client	
Defined	27
SAS Web Client	
defined	27
logs	180
Satellite. See Opware Satellite.	
scripts	
Command Engine	30
deleting backup files	195
Distributed Scripts	
permissions required	301
Secondary Data Access Engine	199
server certificate	
extracting	
Microsoft Active directory from	100
Novell eDirectory from	100
SunDS from	100
importing, external LDAP from	99
server management	
required permissions	250
setting	
customer permissions	83
facility permissions	83
feature permissions	85
Global Shell permissions	87
other feature permissions	87
SAS Client feature permissions	86
server group permissions	84
software provisioning	
overview	48
preview reconcile	49
remediate	50
Software Repository	
defined	27, 33
logs	180
mapping	36
system diagnostic tests	162
Software Repository Cache	
applying, Manual updates	154
defined	28, 34

managing	145
overview	151
packages, availability of	146
staging files	155
Software Repository Multimaster Component	
conflicts	109
defined	28
logs	181
Software Repository Replicator	
configuration	312
defined	27, 33
logs	181
overview	311
prerequisites	311
Software Repository, Multimaster Component	
defined	34
standalone installation	26
supported	
external LDAP directory servers	94
system configuration	
overview	241
required permissions	251
setting configuration parameters	241
system diagnosis	
Command Engine tests	163
contacting, support	158
Data Access Engine tests	161
diagnosing, problems	159
Model Repository Multimaster Component tests	163
running, system diagnosis	164
Software Repository tests	162
testing	160
troubleshooting, problems	158
Web Services Data Access tests	162
system diagnosis, tools	159, 160, 164

T

tools	
content management tools	51
multimaster tools	113, 117
system diagnosis	159, 160, 164
troubleshooting	158

U

uninstalling, patch	45
updating, Microsoft patch	47
uploading, patch	41
user groups	

adding a user	82
adding users to CDR	102
code deployment	308
creating	82
predefined	78
SAS Client permissions	303
SAS Web Client permissions	303
security administrator, process overview	79
setting	
customer permissions	83
facility permissions	83
feature permissions	85
other feature permissions	87
SAS Client feature permissions	86
server group permissions	84
users	
creating	80
deleting	81
editing, user information	81
importing, external LDAP users	101
overview	71
viewing permissions	81
users group	
adding, users to Administrator group	89
removing, users from Administrator group	90
users, of Opsware	23

V

viewing	
facilities information	137
permissions	81
realm information	139
Virtualization Director	
permissions required for	294
Visual Application Manager	
permissions required for	293
visual packager, overview	65

W

Web Services Data Access Engine	
defined	28, 34
logs	181
system diagnostic tests	162
Web Services Data Access Engine, modifying,	
configuration file	95

