Opsware® System 5.1 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

Table of Contents

Preface

Welcome to Opsware System 5.1 – an enterprise-class software solution that enables customers to get all the benefits of Opsware, Inc. data center automation platform and support services. Opsware System 5.1 provides a core foundation for automating formerly manual tasks associated with the deployment, support, and growth of server and server application infrastructure.

About this Guide

This guide describes how to administer Opsware System 5.1, including how to set up users, user groups, administrators, customers, and facilities. It provides information about permissions that need to be set for different levels of operation and access, and it discusses how to monitor and diagnose the health of the Opsware System components.

This guide is intended for Opsware administrators who will create and modify customer accounts, populate drop-down lists, resolve database conflicts in multiple core environments, monitor logs, and stop and restart components.

Contents of this Guide

This guide contains the following chapters:

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

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

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

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

About Opsware Documentation

Conventions in this Guide

This guide uses the following typographical and formatting conventions.

NOTATION	DESCRIPTION
Bold	Defines terms.
Italics	Identifies guide titles and provides emphasis.
`Courier`	Identifies text of displayed messages and other output from Opsware programs or tools.
`Courier Bold`	Identifies user-entered text (commands or information).
`Courier Italics`	Identifies variable user-entered text on the command line or within example files.

Icons in this Guide

This guide uses the following iconographic conventions.

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

Guides in the Documentation Set and Who Should Read Them

- The *Opsware System 5.1 User's Guide* is intended to be read by the system administrator who is responsible for performing the day-to-day functions of managing servers, provisioning operating systems, uploading packages, setting up the Software Tree and node hierarchies, attaching software applications and installing them on servers, managing patches, reconciling servers with software, creating and executing scripts, tracking configuration, and deploying and rolling back code and content. It also documents the day-to-day functions of managing servers, such as server compliance and auditing, software packaging, application configuration, agent deployment, and "global" shell remote data center management.
- The *Opsware System 5.1 Administration Guide* is intended to be read by Opsware administrators who will be responsible for setting up accounts for users, creating user

groups and additional Opsware administrators, assigning permissions for different levels of operation and access, adding customers and facilities, and monitoring and diagnosing the health of the Opsware System components.

- The *Opsware System 5.1 Deployment and Installation Guide* is intended to be used by system administrators who are responsible for the installation of Opsware System in a facility. It documents how to run the Opsware Installer and how to configure each of the components.
- The *Opsware System 5.1 Planning Guide* is intended to be used by advanced system administrators who will be responsible for planning all facets of an Opsware System 5.1 installation and deployment. It documents all the main features of the Opsware system and scopes out the planning tasks necessary to successfully deploy the Opsware System. Sections include: planning the Opsware System 5.1 design for a core, types of installations, and discusses business goals that can be achieved using the software. It also includes information on system sizing, checklists, and best practices.
- The *Opsware System 5.1 Configuration Guide* is intended to be used by system administrators who are responsible for all facets of configuring the Opsware Command Center. It documents how to set up users and groups, configure Opsware server management, and setting up the main Opsware Command Center features, such as patch management, configuration tracking, software repository replicator setup, code deployment, as well as OS and application provisioning.

Contacting Opsware Inc.

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

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

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

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

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

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

The Opsware Customer Support email address and phone number follow:

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

Chapter 1: Opsware System Overview

IN THIS CHAPTER

This chapter provides an overview of the Opsware System 5.1 components and how they interact. This chapter contains the following topics:

- Opsware System Technology
- Opsware System Components
- Interaction Among Opsware System Components

Opsware System Technology

The Opsware System 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.

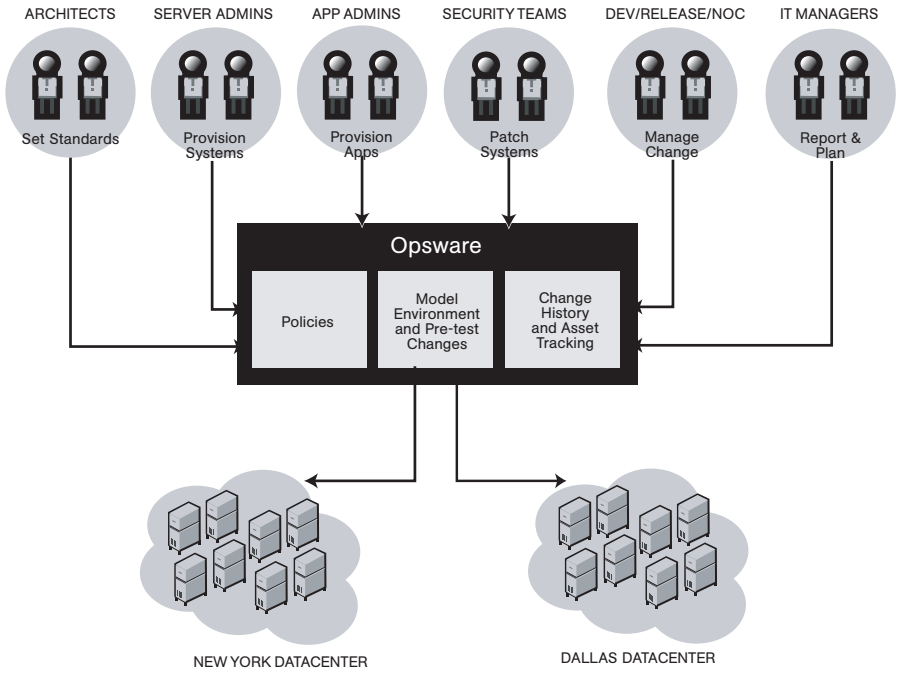
The Opsware System does not just automate your operations, it also allows you to make changes more safely and consistently because you can model and validate changes before you actually commit the changes to a server. Opsware System automation helps ensure that modifications to your servers work the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

Using the Opsware System, 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.

The Opsware System allows you to incorporate and maintain operational knowledge that has often been 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. The ability to store this knowledge in a central repository 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 the Opsware System automates server and application operations across all major platforms and a wide range of applications. Each feature that is shown in the diagram is discussed in the following sections.

Figure 1-1: Overview of Opsware System Features



Types of Opsware Users

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

Table 1-1: Types of Opsware 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 Opsware 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 application provisioning.
Site Engineers and Customer Project Managers	Deploy custom code on servers.

In addition to the Opsware users listed in the above table, 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 Opsware guides, these users are referred to as Opsware users or system administrators. These users log into the Opsware Command Center and OCC Client and use these user interfaces to manage servers in their IT environment.
- **Opsware Administrators** are the users, with special training and information, who are responsible for installing and maintaining the Opsware System. In the Opsware guides, these users are referred to as Opsware administrators. They use the Administration features in the Opsware Command Center to manage the Opsware System and Opsware users (by adding user accounts and assigning permissions for different levels of operation and access), to add customers and facilities, and to change Opsware System configurations. They monitor and diagnose the health of the Opsware System components. Opsware administrators need to understand how Opsware System features operate to support users and the Opsware System.
- **Policy Setters** are the power users who are responsible for architecting what the Opsware System 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 the Opsware System by defining the Software Tree, preparing Operating System Definitions, and acting as Patch Administrators to approve patches for installation in the operational environment.

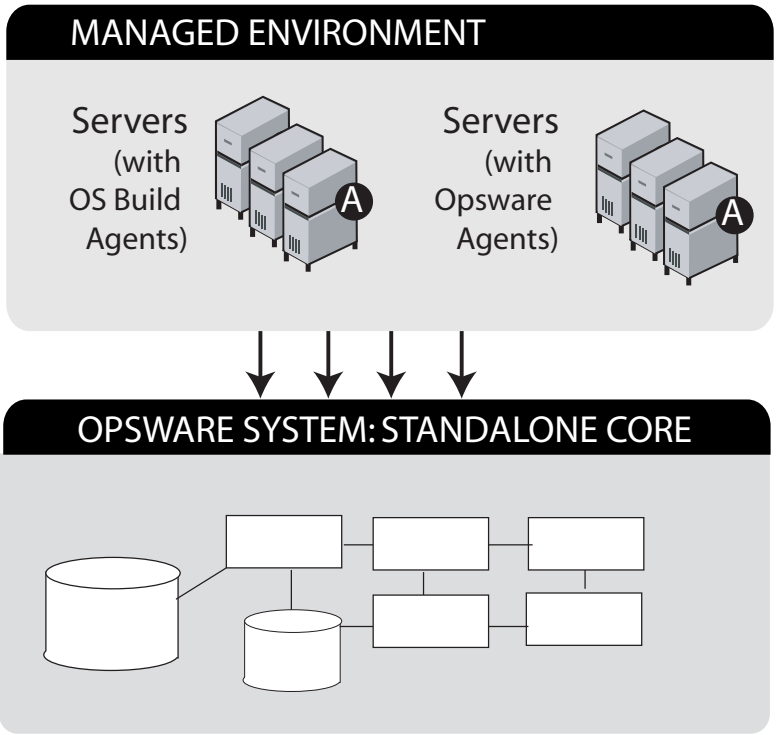
The Opsware System Environment

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

- *Opsware System 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 the Opsware System manages by virtue of the Opsware 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 the Opsware System and guiding the OS installation process. See Figure 1-2.

Figure 1-2: Opsware System Environment



Model-Based Control

The Opsware System utilizes a model-based control approach to accomplish infrastructure management.

Users and administrators interact with the Opsware Command Center, a Web-based front-end application, to accomplish Opsware System tasks such as server management, software distribution, patch management and installation, inventory reporting, system diagnosis, and code and content deployment to the operational environment. The Opsware System 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 Opsware System components work from, or update, a data model of information maintained in the Model Repository for all servers that the Opsware System manages.

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

Types of Opsware System Installations

There are three basic types of Opsware System 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 Opsware Satellites.) A core contains all components of the Opsware System, except for the Opsware 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 Opsware System data at each facility.
- **Satellite** – Installed in a remote facility, an Opsware 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 Opsware core or Satellite is associated with a specific facility.

Opsware System Components

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

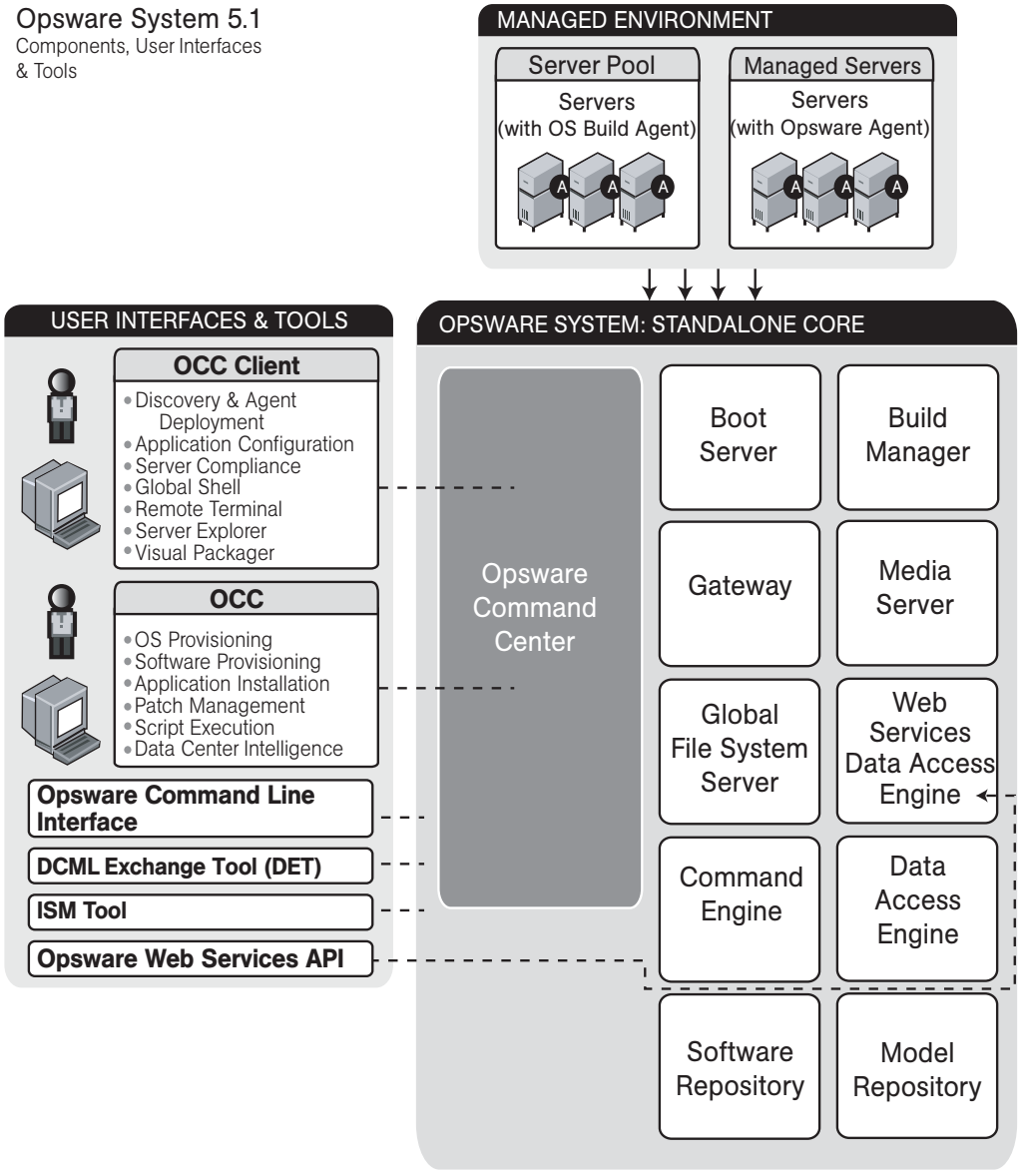
The sections that follow describe the components of the Opsware System:

- **Boot Server** – Part of the OS Provisioning feature, supports network booting of Sun and x86 systems.
- **Build Manager** – Facilitates communication between components for OS provisioning.
- **Command Engine** – A system for running distributed programs across many servers.
- **Data Access Engine** – An XML-RPC interface to the Model Repository.
- **Media Server** – Provides network access to vendor-supplied media used during OS provisioning.
- **Model Repository** – The Opsware System's data repository (database).
- **Model Repository Multimaster Component***t* – The application that propagates and synchronizes changes from each Model Repository database to all other Model Repository databases.
- **Opsware Agents** – An intelligent agent that runs on each server that the Opsware System manages.
- **Opsware Command Center** – A user interface to the Opsware System.
- **OS Build Agent** – Responsible for registering a bare metal server with the Opsware System and guiding the OS installation process.
- **Software Repository** – The central repository for all software that the Opsware System manages.
- **Software Repository Replicator** – 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** – 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** - Contains local copies in the Opsware Satellite of the the Software Repository of the core (or another Satellite).
- **Web Services Data Access Engine** – Provides increased performance from the Model Repository to other Opsware System components.
- **Opsware Gateway** – Provides network connectivity to Opsware cores and Satellites.
- **Global File System Server** – Dynamically constructs the Opsware Global File System (OGFS), a virtual file system.

The following figure shows an overview of Opsware System 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 Boot 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 Opsware Agents). Command Engine scripts are written in Python and run on the Command Engine server. Command Engine scripts can make Remote Procedure Calls (RPC) on Opsware Agents. These calls are delivered in a secure manner and are auditable by using data stored in the Model Repository.

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

Because interactions with the Model Repository go through the Data Access Engine, clients are less impacted by changes to the Model Repository's schema. The Data Access Engine allows features to be added to the Opsware System 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 System components work from, or update, a data model maintained for all servers that the Opsware System 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 the Opsware System manages has an intelligent agent running on that server. The Opsware Agent is the agent of change on a server. Whenever the Opsware System needs to make changes to servers, it does so by sending requests to the

Opsware Agent. Depending on the request, the Opsware Agent might use global Opsware System services (such as the Model Repository and Software Repository) in order to fulfill the request. Some functions that the Opsware Agent supports are:

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

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

Dormant Opsware Agents

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

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

Opsware Command Center

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

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

OS Build Agent

The OS Build Agent, part of the OS Provisioning feature, is responsible for registering bare metal servers in the Opsware System. In addition, it is the agent of change on the server during the OS installation process (that the Build Manager manages) through to the point at which the actual Opsware Agent is installed.

Software Repository

The Software Repository is the central repository for all software that the Opsware System 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 Opsware Agent can install software running on the server where the Opsware Agent is installed. The Model Repository then updates its record of the software installed on the server. This process of updating the actual software configuration of a server with a specified configuration stored in the Model Repository is called reconciliation.

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

See the *Opsware System 5.1 Configuration Guide* for information about how to upload software packages to the Software Repository.

Software Repository Replicator

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

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

Software Repository Cache

Installed in an Opsware 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, if a Solaris package that resides on Software Repository (A) is needed for installation in a second facility that contains Software Repository (B) that is part of the same multimaster mesh, the Multimaster Component allows B to discover the presence of the package on A. The package is then transferred and cached at B so that it can be used in the second facility.

Web Services Data Access Engine

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

Opsware Gateway

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

Global File System Server

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

Interaction Among Opsware System Components

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

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

General Interaction Among Components

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

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

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

Opsware System Security

To enable secure communication with the Opsware Agent, the Opsware System 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 Opsware Agent to establish a secure https connection to the Opsware System components.

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

OS Provisioning

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

Because the OS Provisioning feature supports installation-based provisioning, your organization can keep its OS installations lean. Rather than trying to manage changing software through master images, you can use the OS Provisioning feature to install and remove often changing software, including system patches, system utilities, and third-party agents (such as monitoring, backup, and anti-viral agents). Figure 1-4 shows the OS Provisioning feature interaction.

See the *Opsware System 5.1 User's Guide* for information about the OS provisioning process.

Figure 1-4: OS Provisioning Subsystem Interaction

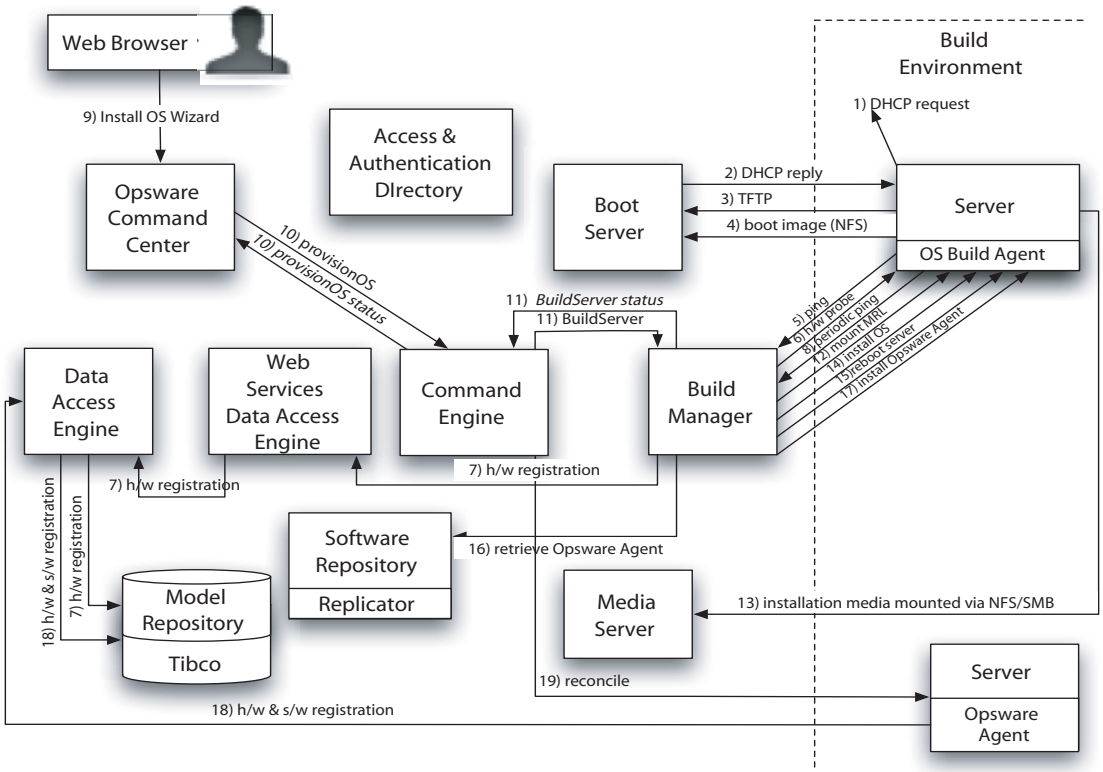


Figure 1-4 shows the following transactions taking place:

1 The DHCP request is a network broadcast.

2 The DHCP reply contains the IP address of the Build Manager for use by Sun Solaris and Red Hat Linux provisioning. For Microsoft Windows provisioning, the DNS configuration in the DHCP reply must resolve the hostname *buildmgr*.

3 TFTP is used to boot the server over the network (using inetboot for Solaris, and PXE for Windows and Linux). Windows and Linux can use a boot floppy instead of PXE.

4 An NFsboot 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 hardware detected in Step 6 is then used to register the server with the Opsware System.

8 The OS Build Agent periodically contacts the Build Manager with a ping message. The system remains in this state unless a user begins a provisioning process with the Opsware Command Center, or the server is removed from the network, in which case it will eventually be removed from the Server Pool.

9 A user initiates OS provisioning with the Install OS Wizard in the Opsware Command Center.

10 **11** Feedback is provided throughout OS provisioning with status messages passed from the Build Manager to the Command Engine and from the Command Engine to the Opsware Command Center.

12 A Media Resource Locator contains the network location (hostname and path) of an NFS or SMB server from which to retrieve the vendor OS installation media.

13 Solaris and Linux provisioning use NFS. Windows provisioning uses SMB.

14 The vendor installation program is used to install the OS (Sun Solaris Jumpstart, Red Hat Linux Kickstart, or Windows unattended.txt).

15 The server is rebooted after OS installation.

16 **17** The OS Build Agent is used to install the Opsware Agent.

18 Hardware and software registration is performed as part of the Opsware Agent installation.

19 The Reconcile function installs additional software that the vendor installation program did not install.

Steps 12 through 18 are managed by a build script that runs inside the Build Manager. The build script is invoked by provisionOS and manages the OS installation at a micro level. ProvisionOS is a Command Engine script that is responsible for managing the installation process at a macro level.

Patch Management

The Opsware System 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 1-5 through Figure 1-8.

See the *Opsware System 5.1 User's Guide* for information about the patch management process.

Figure 1-5: Patch Management Feature: Upload Patch

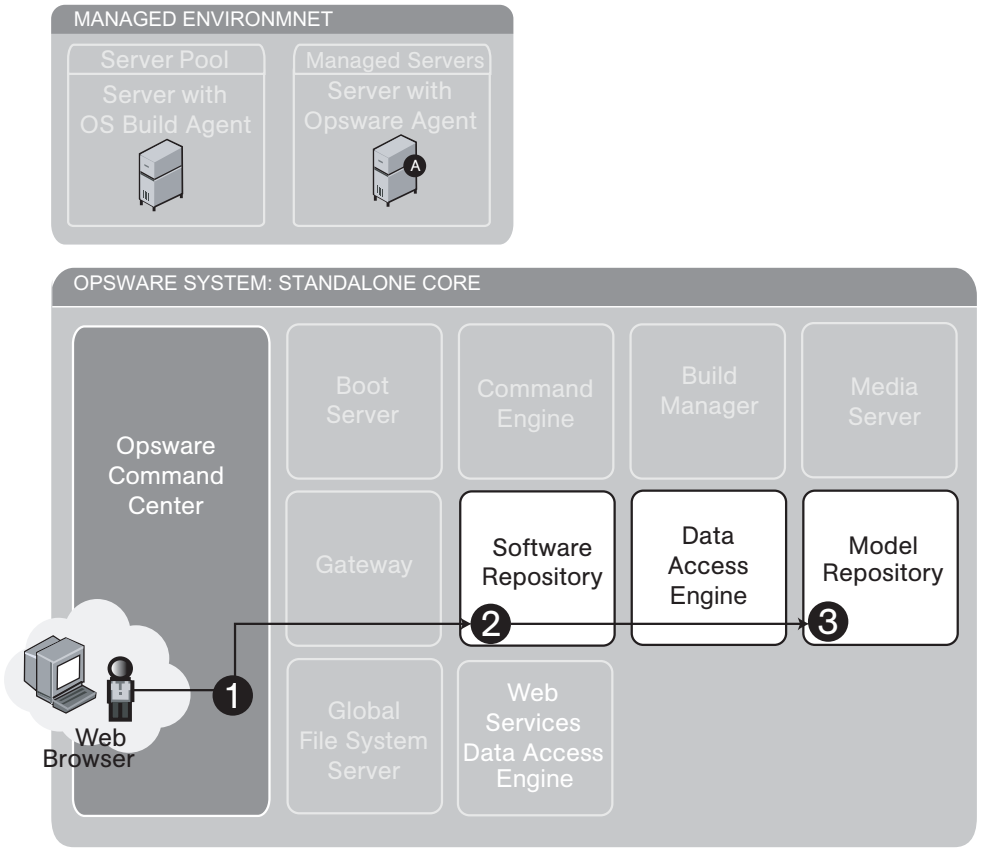


Figure 1-5 illustrates the following upload patch process:

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

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

Figure 1-6: Patch Management Feature: Install Patch

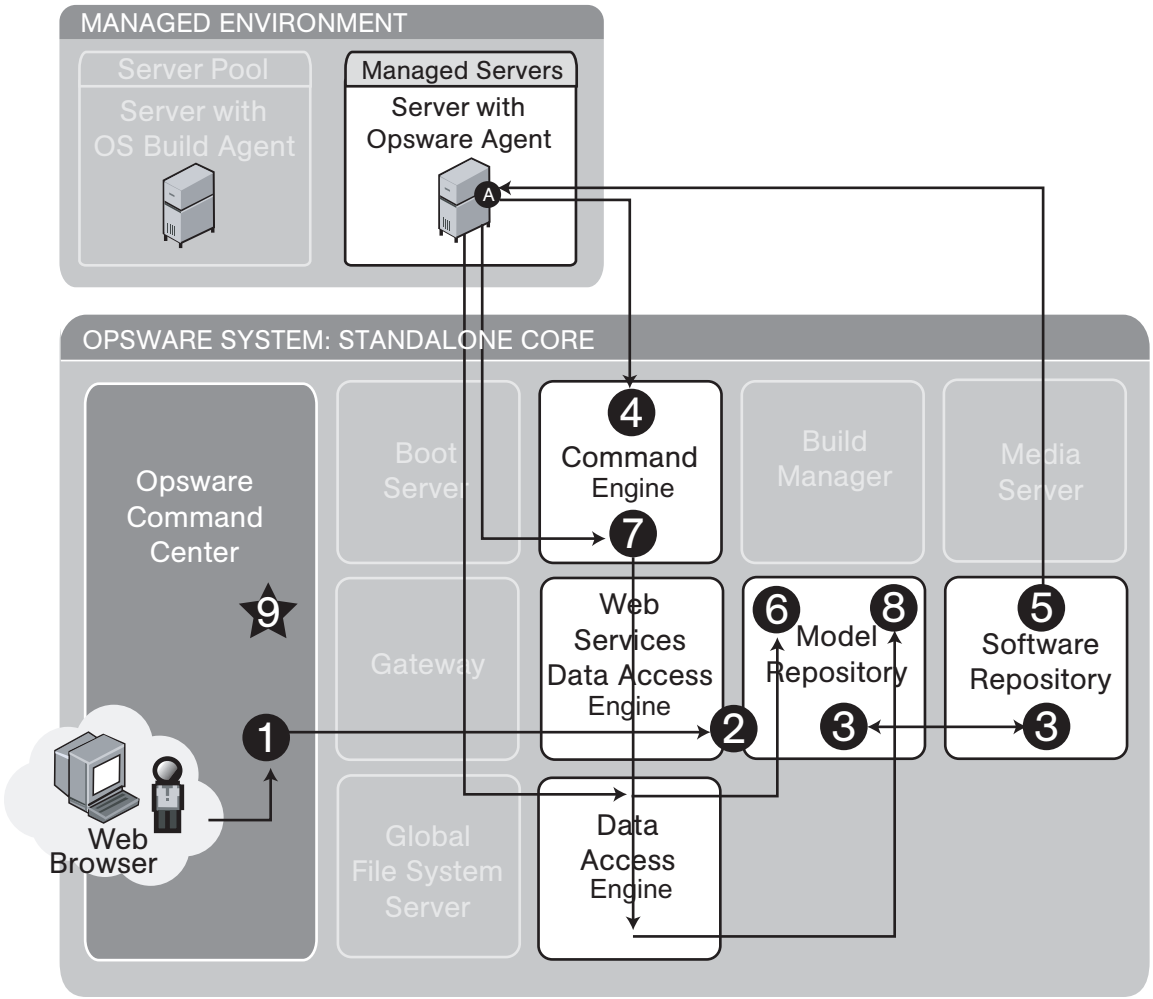


Figure 1-6 illustrates the install patch process:

1 An Opsware user with the required permissions logs into the Opsware Command Center and clicks the Install Patch wizard link on the Home page.

2 Using the wizard, the user specifies patches and servers and starts the install process, retrieving patch information from the Model Repository via the Web Services Data Access Engine.

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

Figure 1-7: Patch Management Feature: Uninstall Patch

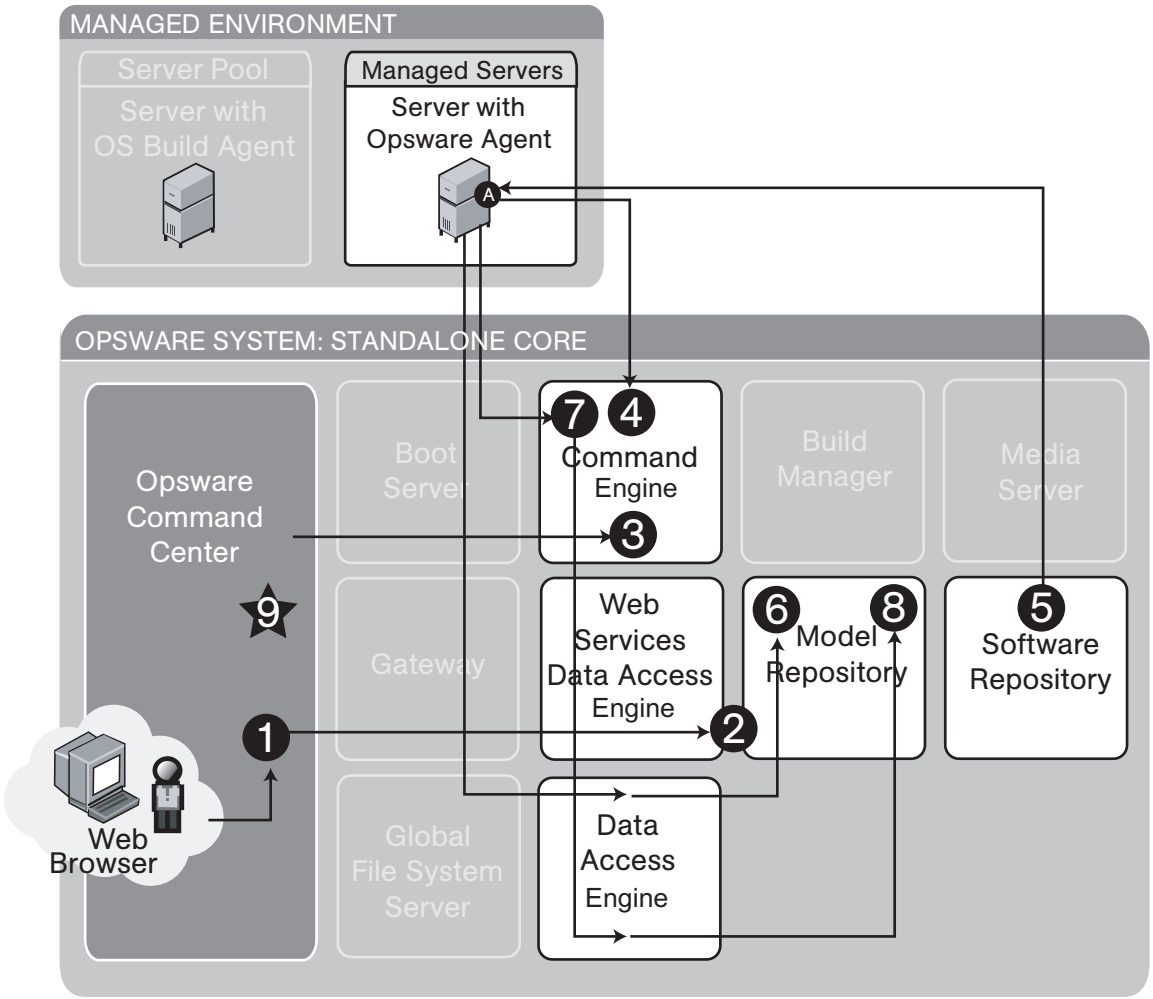


Figure 1-7 illustrates the uninstall patch process:

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

4. The Command Engine gets a list of installed software from the Opsware 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 Opsware Agent on each managed server removes the patch from the managed servers and performs all required uninstall operations and reboots.
6. When uninstallation is complete, a record of all currently-installed software is stored in the Model Repository via the Data Access Engine.
7. The Opsware Agent on each managed server reports uninstallation status to the Command Engine.
8. The Command Engine stores uninstallation status in the Model Repository via the Data Access Engine.
9. Operation complete status displays in the Opsware Command Center.

Figure 1-8: Patch Management Feature: Microsoft Patch Update

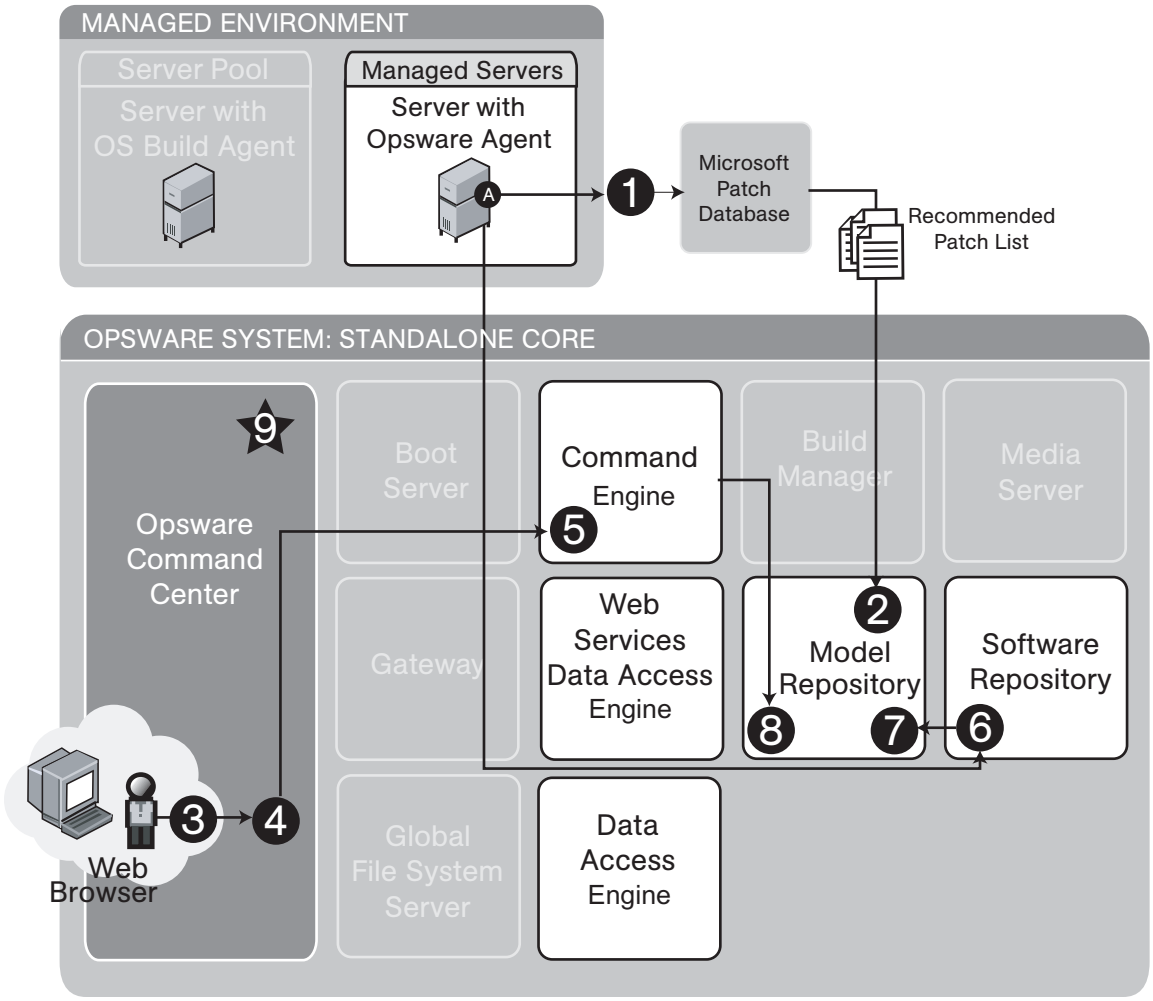


Figure 1-8 illustrates the Microsoft patch update process:

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

4. Using the wizard, the user selects the Windows servers, selects some or all of the patches on the Recommended Patch List, then clicks the Install button.
5. The installation details are passed from the OCC to the Command Engine, which obtains a list of installed software from the Opsware Agent. It compares this list to the user-selected list and determines what actually needs to be installed.
6. Opsware Agent on the managed server downloads patches from the Software Repository and installs them, performing all required install operations and reboots.
7. When installation is complete, a record of all currently installed software is stored in the Model Repository via the Data Access Engine.
8. Install operation status is reported to the Command Engine, which places it in the Model Repository via the Data Access Engine.
9. Operation complete status displays in the Opsware Command Center.

Application Provisioning

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

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

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

See Figure 1-9 and Figure 1-10. See the *Opsware System 5.1 User's Guide* for information about the application provisioning process.

Figure 1-9: Application Provisioning Step 1: Preview Reconcile

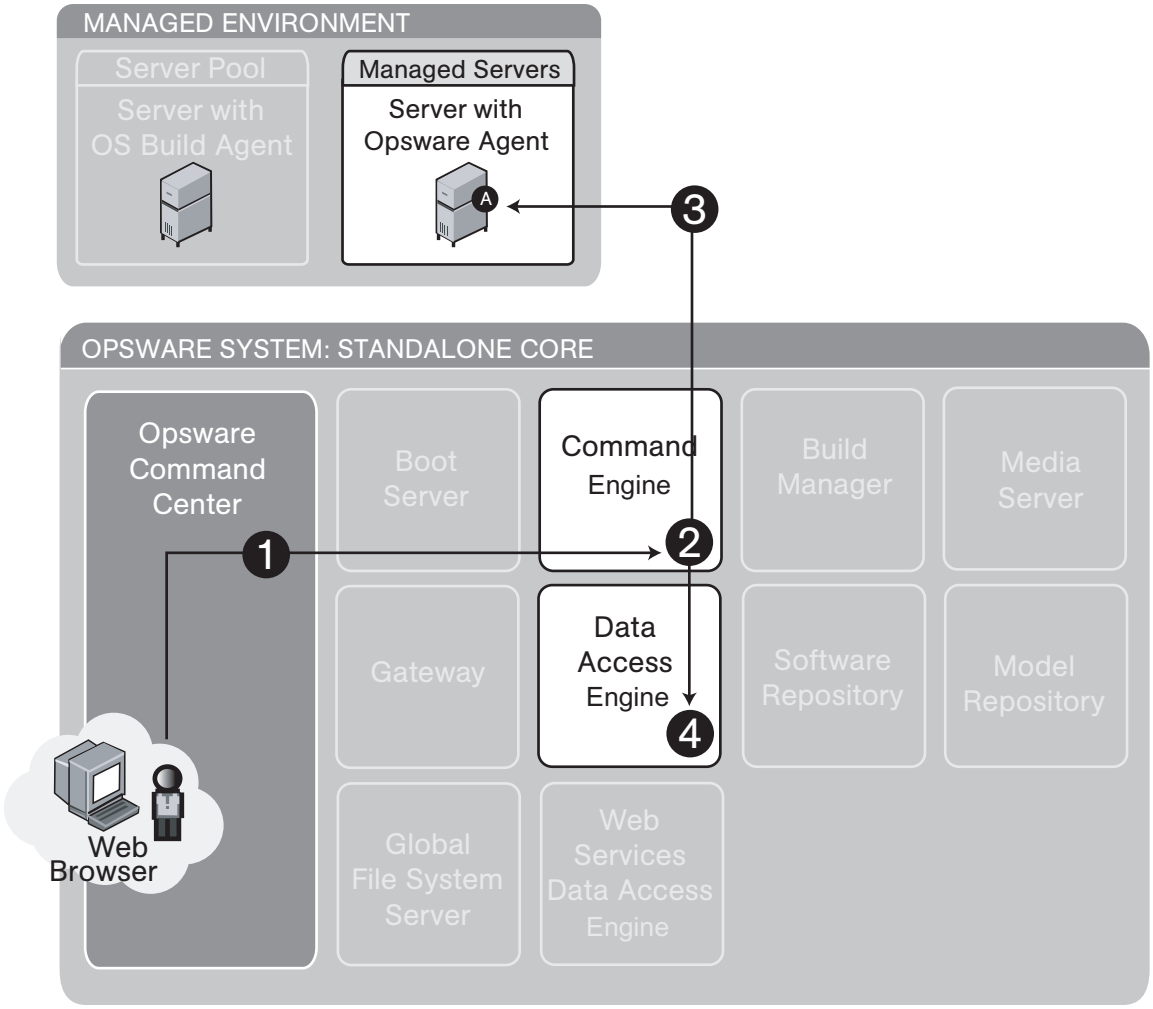


Figure 1-10: Application Provisioning Step 2: Software Installation and/or Removal Through Reconcile

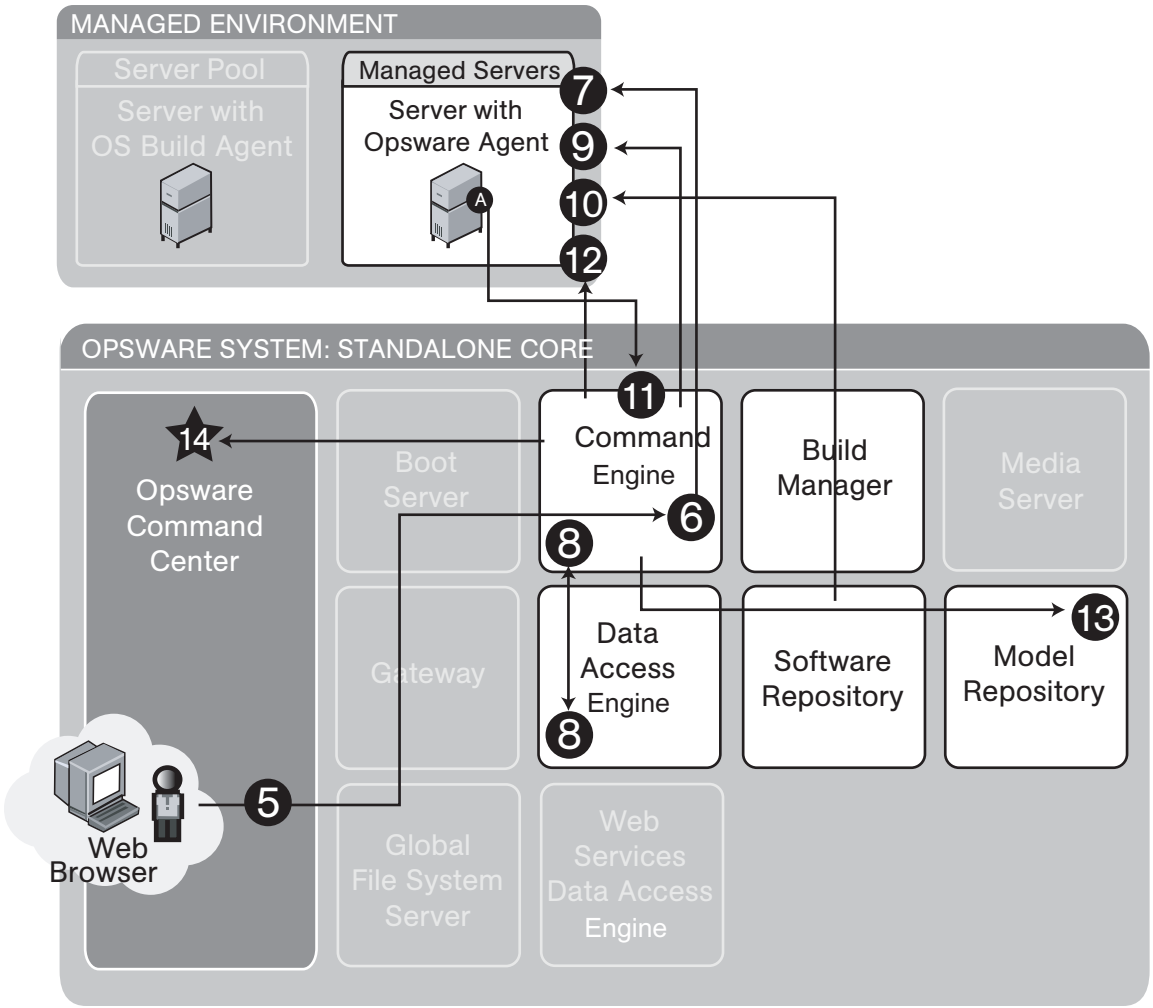


Figure 1-9 and Figure 1-10 illustrate the application provisioning process:

Application Provisioning Step 1: Determine Server Configuration:

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

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

Application Provisioning Step 2: Software Installation through Reconcile

5 At the end of Reconcile Preview, the OCC displays a list of the software to be installed and/or removed. The user confirms proceeding with the reconcile.

6 The OCC starts the Reconcile job in the Command Engine.

7 The Opsware Agent on each of the specified servers is queried by the Command Engine for a list of packages installed on its server.

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

9 The Command Engine tells the Opsware Agent to install and/or remove packages.

10 The Opsware Agent downloads packages from the Software Repository, removes any packages that need to be removed, and installs the new packages, performing all necessary install, uninstall, and reboots, if required.

11 The Opsware Agent reports installation status to the Command Engine.

12 The Opsware Agent on each of the specified servers is queried by the Command Engine for a list of packages installed on its server to confirm what was installed and/or removed.

13 The Command Engine stores installation status in the Model Repository via the Data Access Engine.

14 Status of completed installation and removal of packages displays in the OCC via the Command Engine.

Code Deployment and Rollback

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

After you upload the files and test your changes, you can synchronize updates to the production hosts that run your operational environment. You can run specific synchronizations and perform other service deployment operations by selecting CDR menu options available from the Opsware Command Center navigation panel. Figure 1-11 shows the code deployment and rollback process.

See the *Opsware System 5.1 User's Guide* for information about the process to deploy code and content to servers in the managed environment.

Figure 1-11: Code Deployment and Rollback Feature

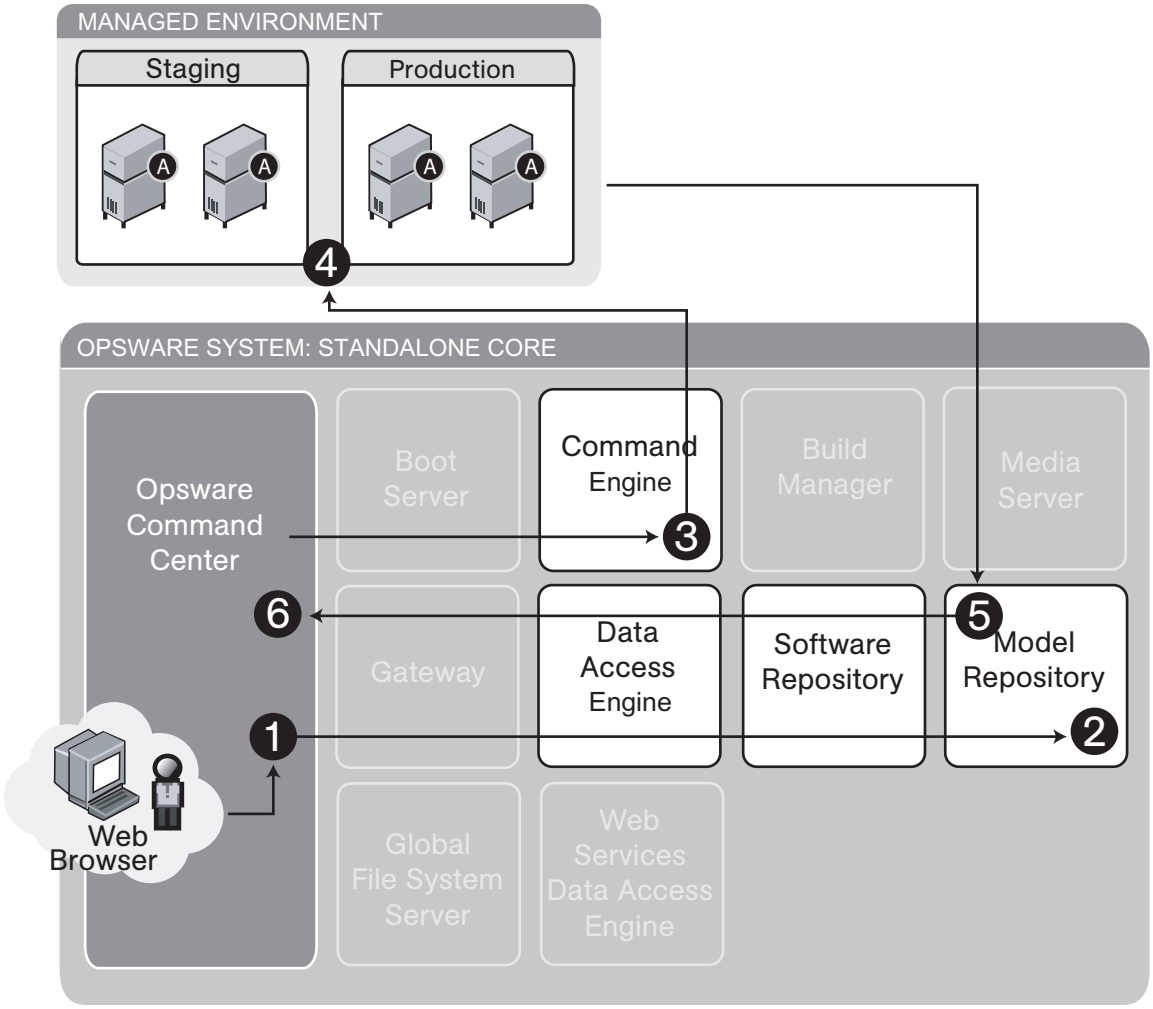


Figure 1-11 illustrates the code deployment and rollback process:

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

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

Script Execution

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

From the Opsware Command Center, you can create or upload a script, set it up to run simultaneously across multiple Unix or Windows servers, and monitor it as it executes on each server. After a script runs, job- and server-specific execution results are available for review. You can modify, delete, or rerun a script at a later date. See Figure 1-12 and Figure 1-13.

See the *Opsware System 5.1 User's Guide* for information about the process to create and execute scripts in the managed environment.

Figure 1-12: Scripting Feature: Upload Script

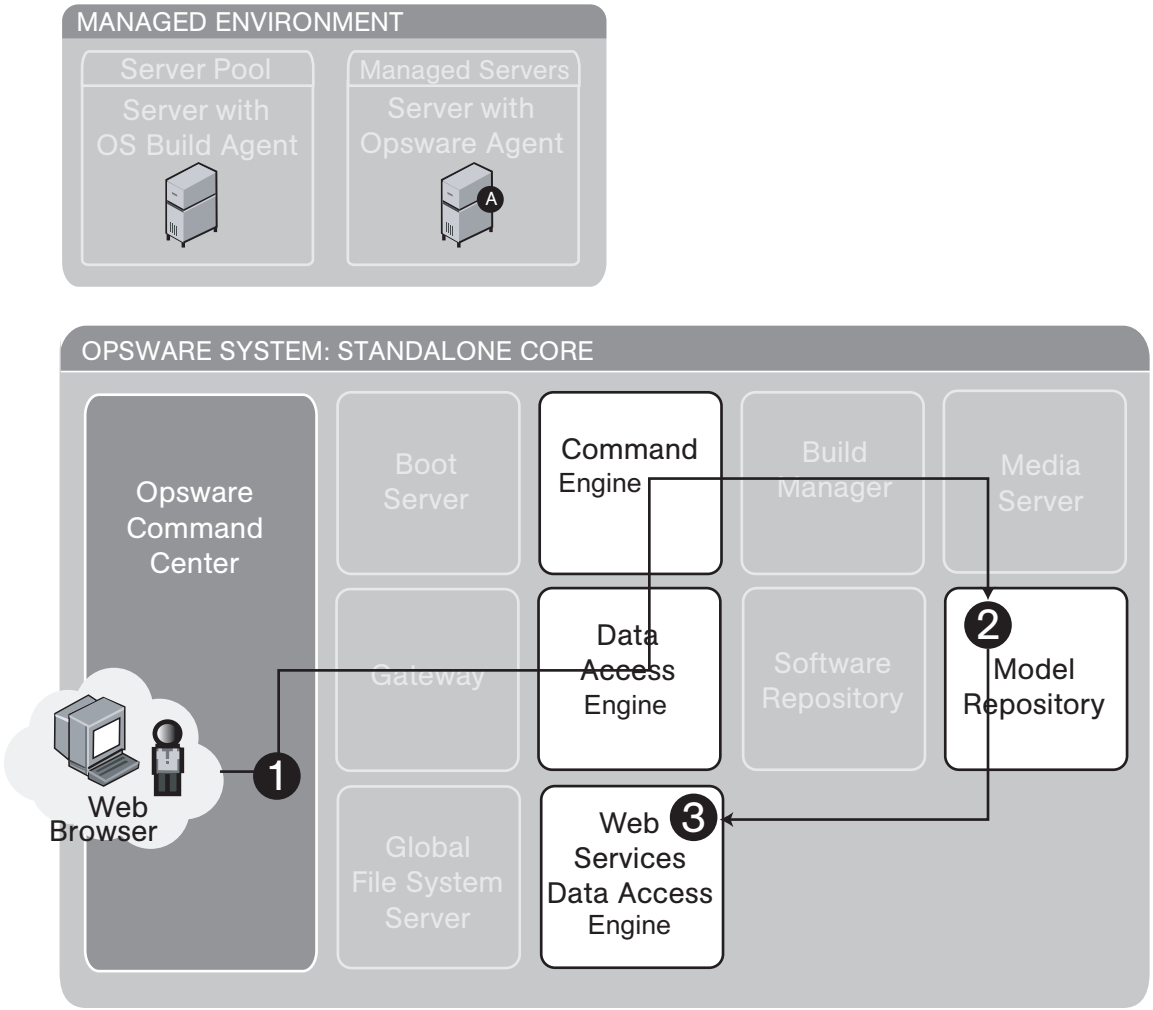


Figure 1-12 illustrates the script execution – upload script process:

1. An Opsware user with the required permissions logs into the Opsware Command Center and clicks the Scripts link under Software and then clicks New Script.
2. The user selects 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 scripted in the list of available scripts.

Figure 1-13: Scripting Feature: Execute Script

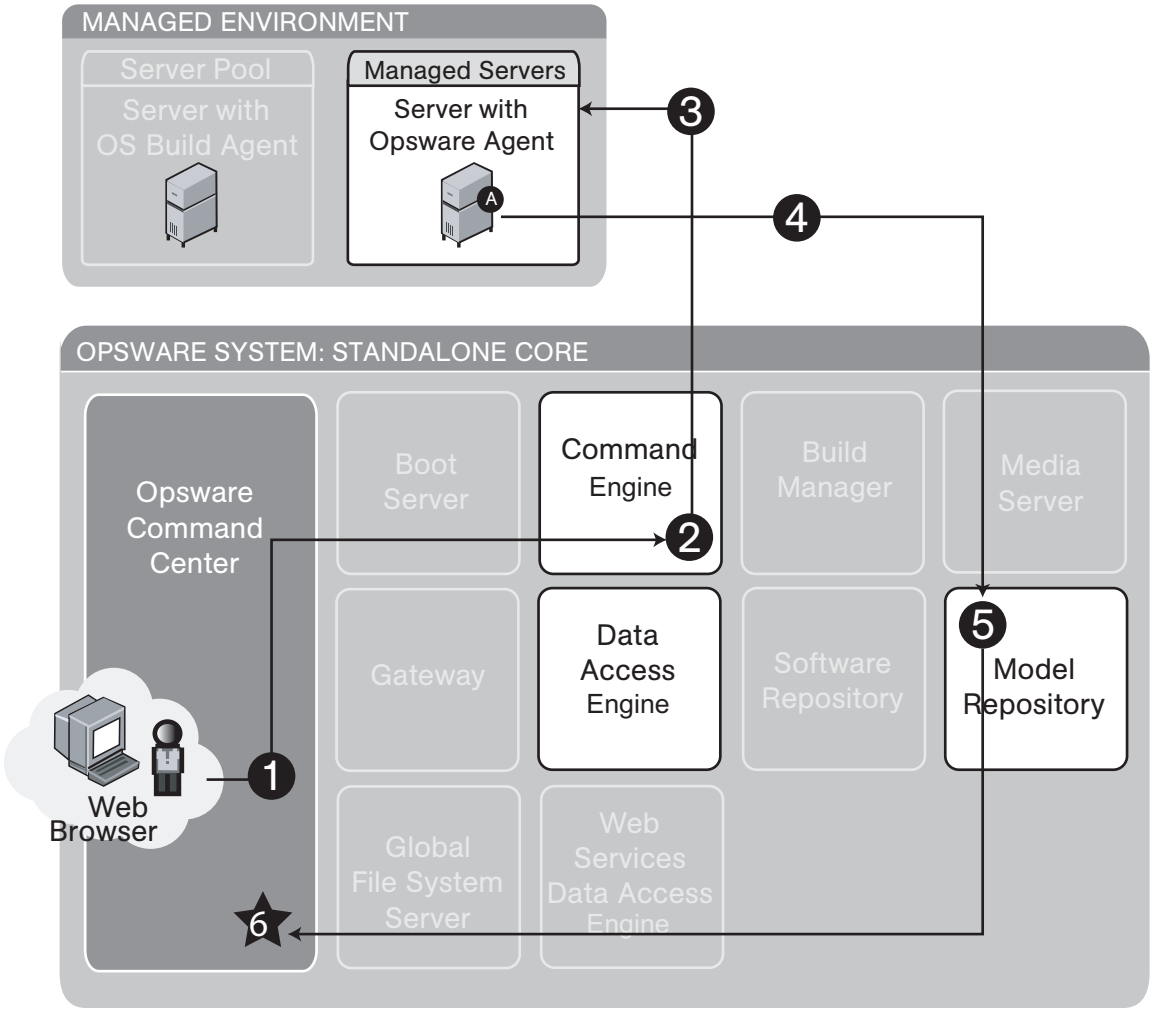


Figure 1-13 illustrates the scrip execution – execute script process:

1 An Opsware user with the required permissions logs into the Opsware Command Center and clicks the Run Distributed Script Wizard link on the Home page.

2 The user selects the scripts and the servers on which to execute the script and clicks Run Script. The request is passed to the Command Engine.

3. The Command Engine contacts the Opsware Agent on the selected servers and tells it to execute the script.
4. The Opsware 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 Opsware Command Center via the Data Access Engine for the user to view.

Integration with AIX and HP-UX Installation Technology

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

See the *Opsware System 5.1 Configuration Guide* for information about how the Opsware System integrates with third-party OS installation technologies.

Figure 1-14 explains the interaction between the Opsware System components when the Opsware System is integrated with AIX NIM and HP-UX Ignite OS installation technologies. Opsware System installation integration with AIX NIM and HP-UX Ignite occurs with the integration of the Opsware Installer. See the *Opsware System 5.1 Configuration Guide* for more information.

Figure 1-14: Opsware Integration with AIX and HP-UX OS Installation Technology

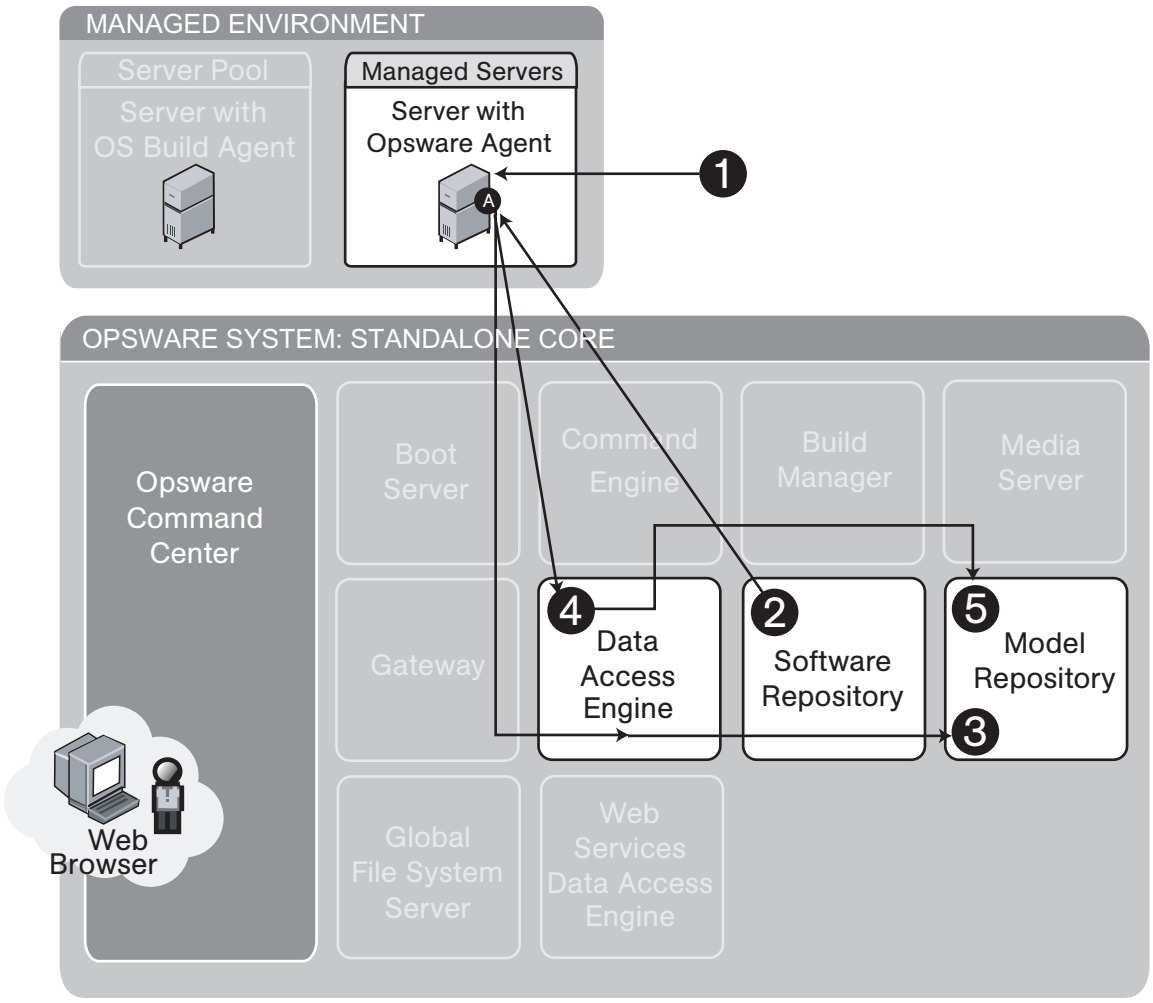


Figure 1-14 illustrates the Opsware System integration with AIX and HP-UX operating systems:

1 Installation technology installs the OS.

2 Opsware System integration downloads and installs the Opsware Agent on the server.

3 The Opsware Agent determines hardware, software, customer, and facility information and records the server information in the Model Repository via the Data Access Engine.

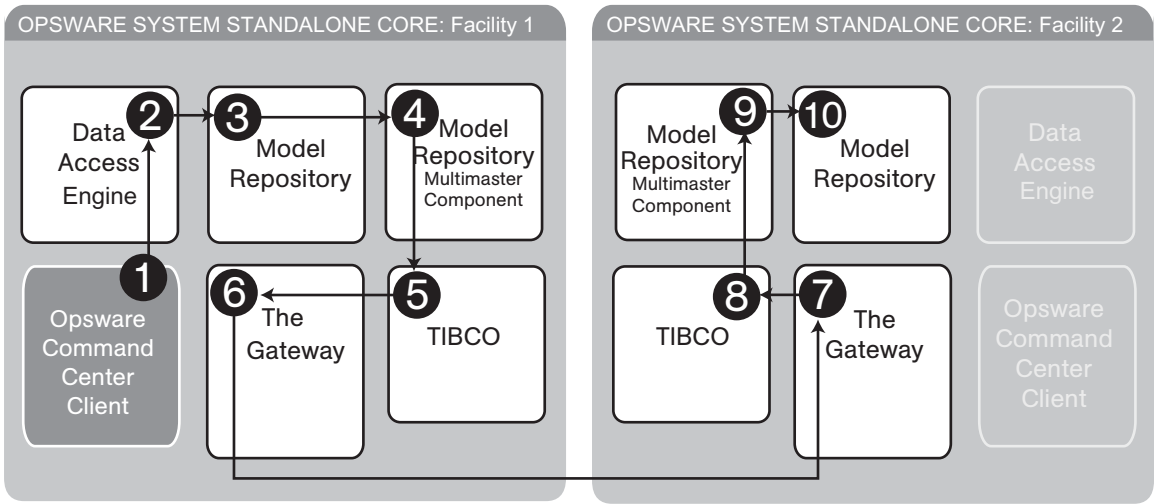
4 The Opsware Agent Installer attaches the server to the specified OS template.

5 (Optional) The server is reconciled with the modeled OS in the Model Repository.

Component Interaction in Multiple Facilities

Figure 1-15 show how the Opsware System components interact when the Opsware System is running in multiple facilities. See “Opsware Multimaster Mesh Overview” on page 49 in Chapter 2 for information about how to administer this Opsware System configuration.

Figure 1-15: Interaction Between Components in Multiple Facilities



1 An Opsware 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 Opsware Gateway in Facility 1 and sends the updated message.

7 The updated message travels over the tunnel between facilities and arrives at the Opsware Gateway in Facility 2.

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

Figure 1-16: Interaction of Discovering Servers and Installing Agents

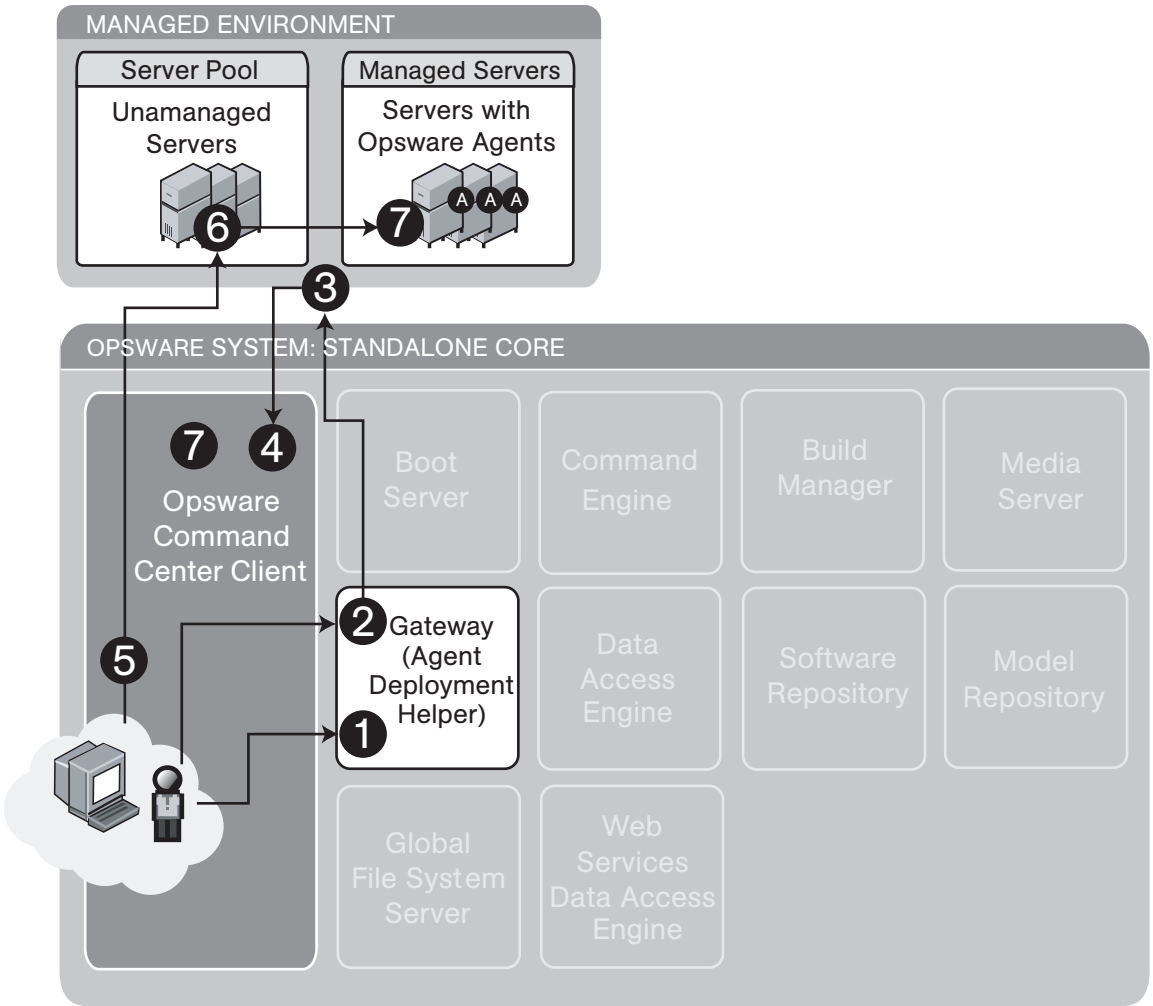


Figure 1-16 shows the process of discovering unmanaged servers and installing the Opsware Agent on those servers:

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

Agent Deployment Helper.

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

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

    For Windows:

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

Application Configuration Management

Opsware 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 can 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 1-17: Application Configuration Management Process

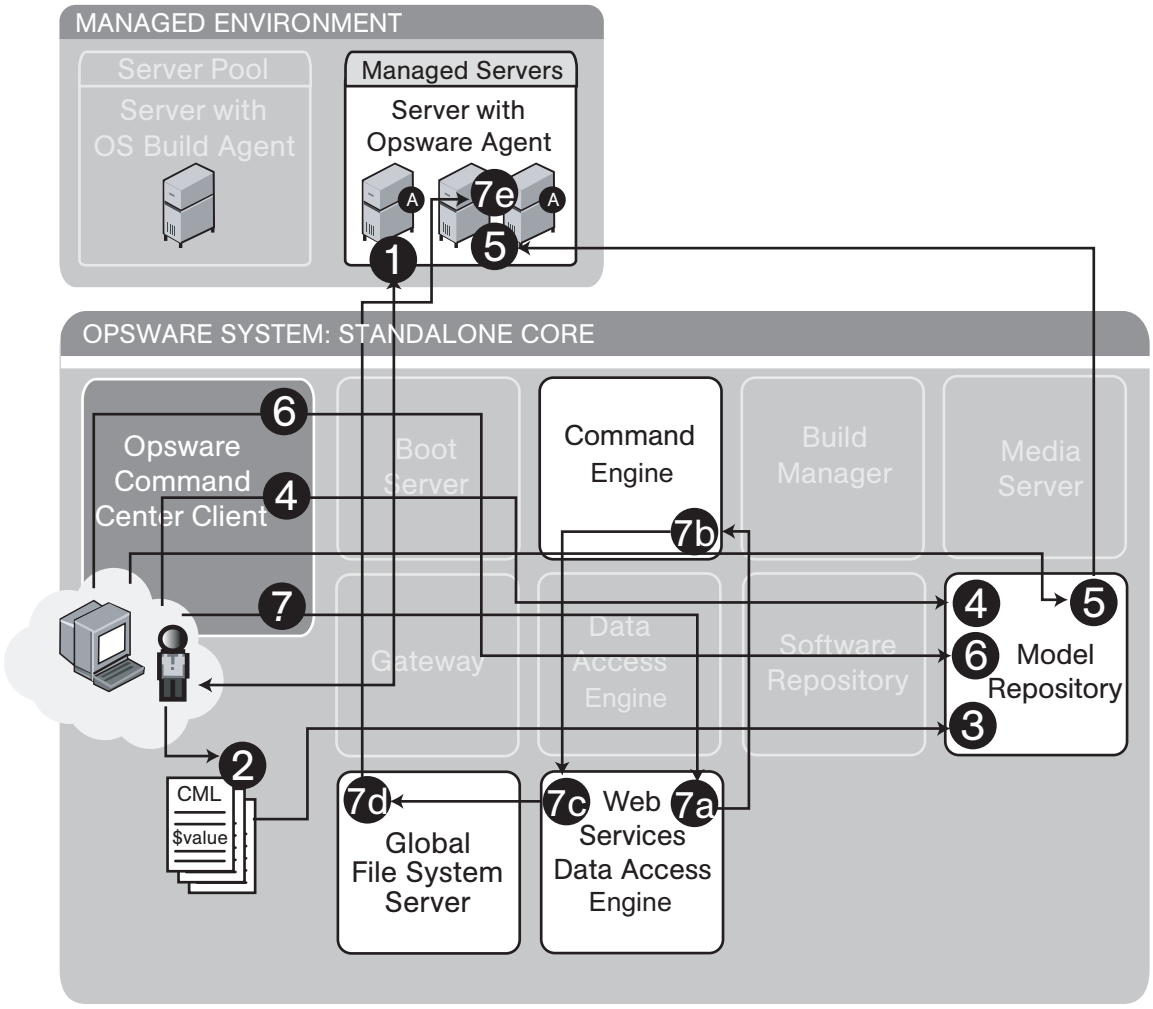


Figure 1-17 shows the process of discovering unmanaged servers and installing the Opsware Agent on those servers:

Part A: Create an Application Configuration and Associated Templates

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

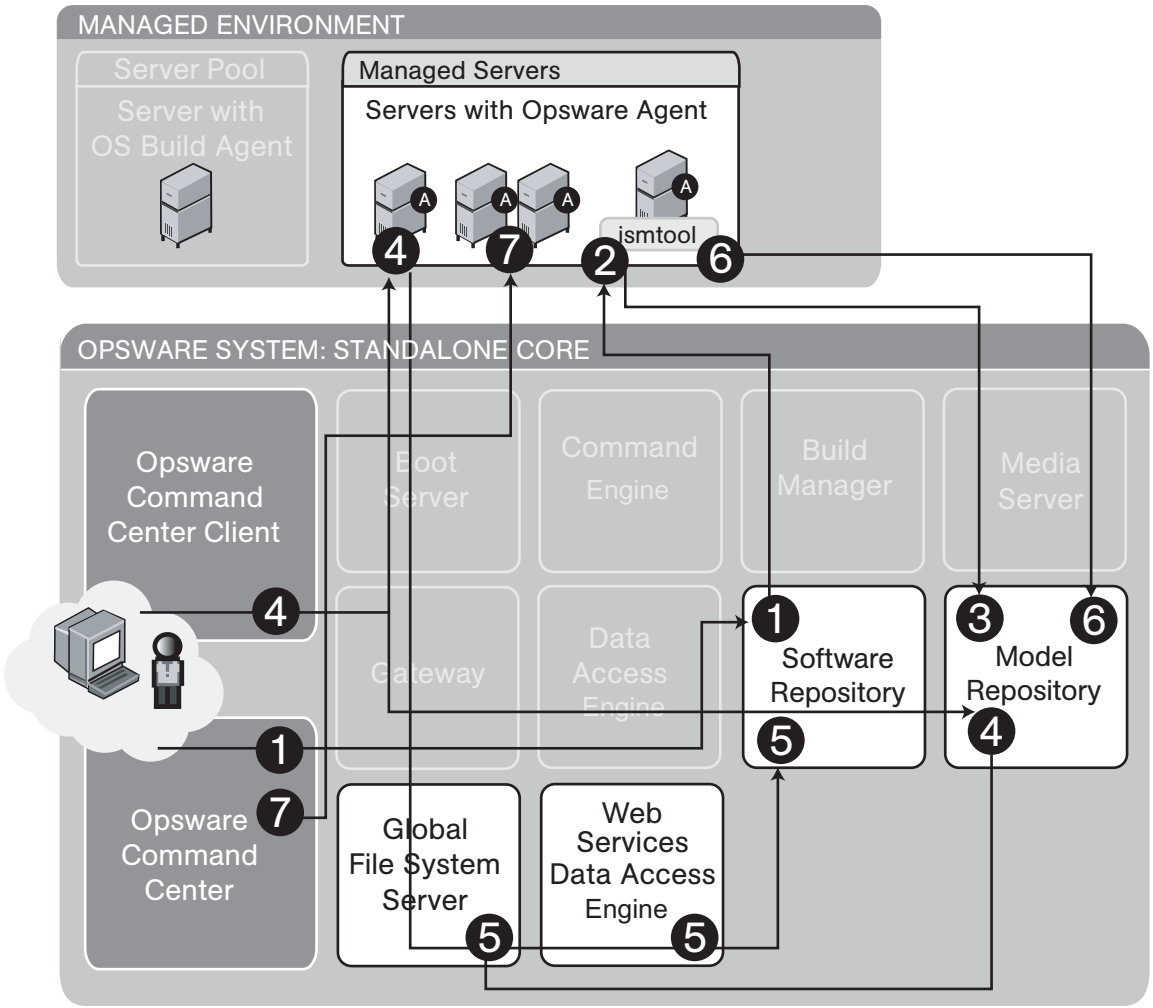
Part B: Configure and Push Application Configurations to Servers

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

Visual Packager

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

Figure 1-18: Interaction of Configuring the Packaging Server and Creating a Package



Part A: Configuring the Visual Packaging Server

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

2. The user selects “Install ➤ By Template...” to install the template.

   Opsware Tools ➤ Visual Packager ➤ [OS of server selected]

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

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

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

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

Part B: Creating and Installing a Visual Package

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

   - A server (from managed servers)
   - An Audit Result (from the Model Respository)
   - A Snapshot Result (from the Model Respository)

2. 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.

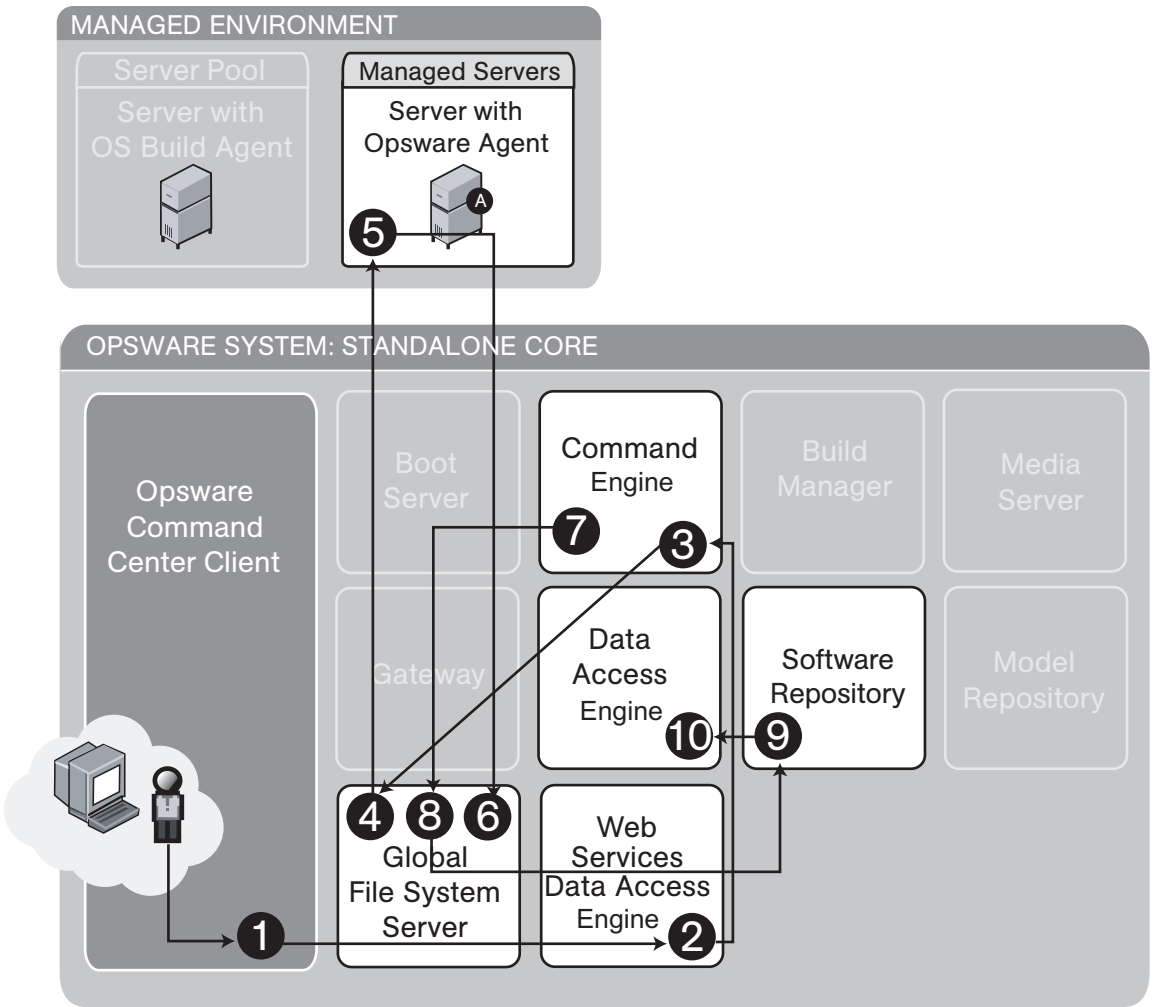
3. The ismtool creates a node in the Software Tree on the model and attaches the new package.

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

Server Compliance and Auditing

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

Figure 1-19: Component Interaction of Auditing a Server



Server Compliance: Take a Snapshot

1. An Opsware user chooses a Snapshot template to use.
2. The user selects 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.

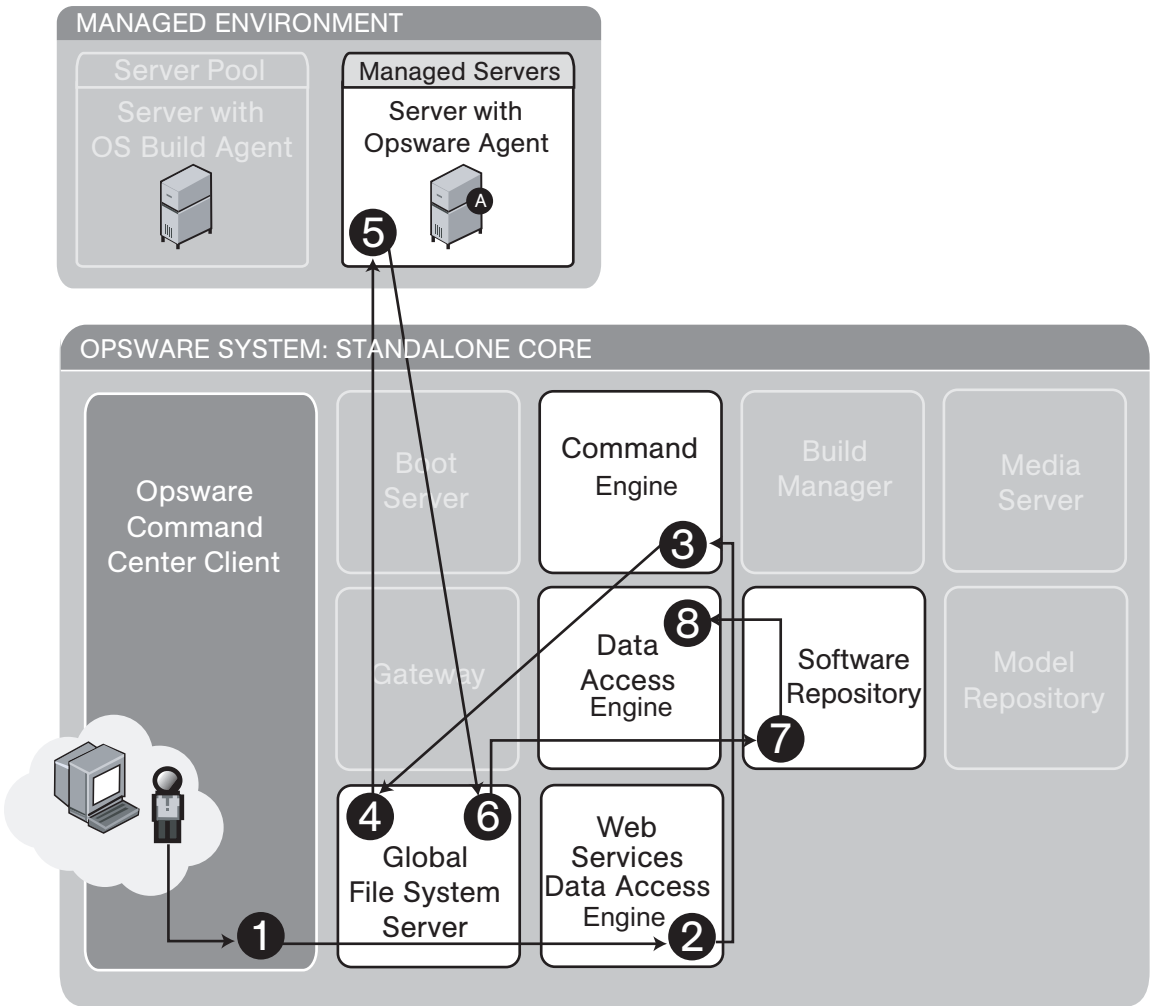
4 The Global File System Server is used to provide snapshot information from the managed server.

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 1-20: Component Interaction of Taking Snapshots

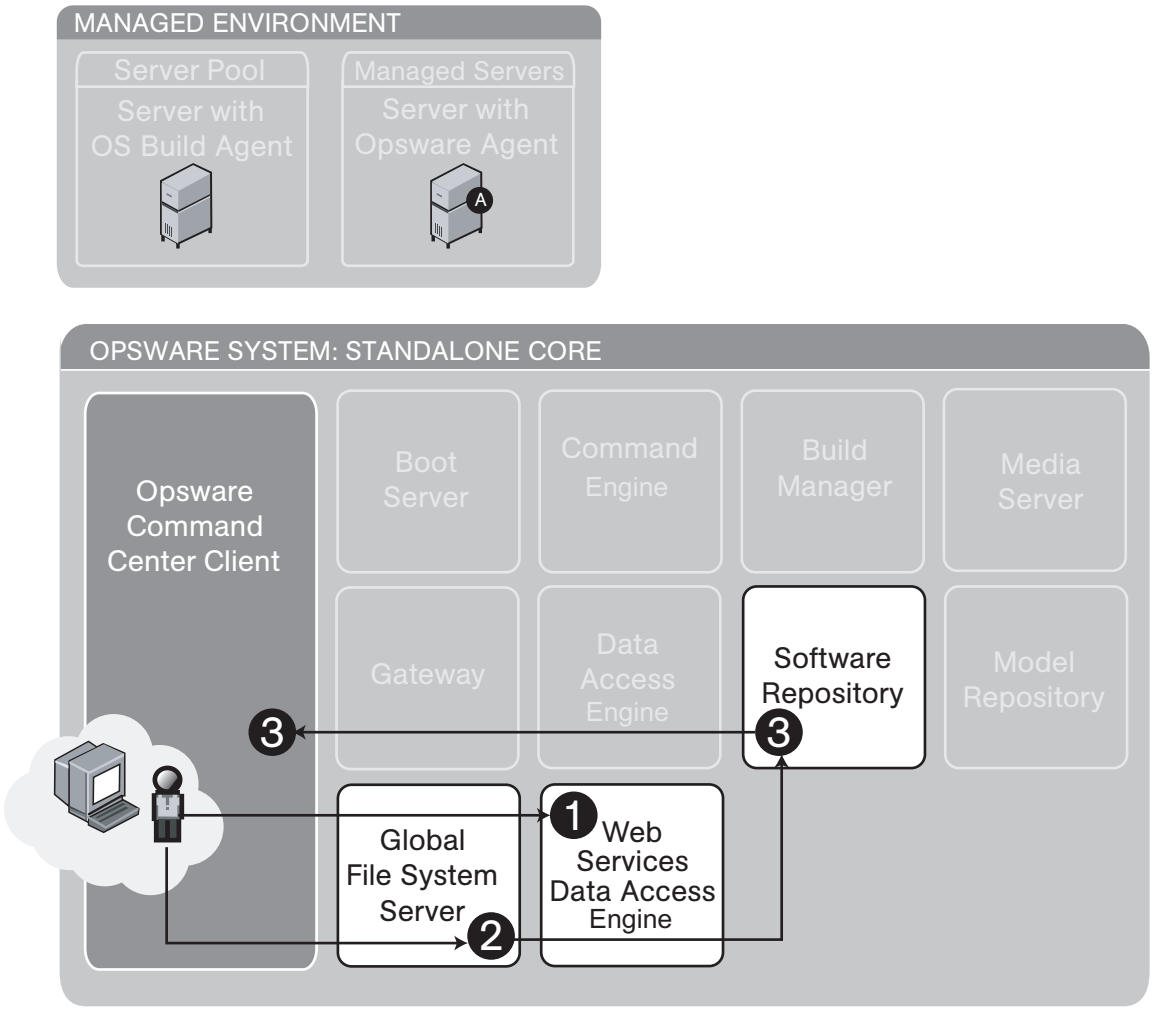


Server Compliance: Run an Audit

1. An Opsware user chooses an audit template to use.
2. The user selects 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...
5. ... from the managed server.
6. The audit information is assembled in the Global File System Server.
7. The Command Engine issues the create audit command.
8. The Global File System Server loads appropriate snapshots and performs a difference.
9. The resulting audit is uploaded to the Software Repository.

10 The audit is stored in the Data Access Engine.

Figure 1-21: Component Interaction of Viewing Snapshot or Audit Results



Server Compliance: View results of audit or snapshot

1 An Opsware user gets list of available snapshots or audit results information.

2 The user requests detailed information about a snapshot or audit.

3 The results are returned from the Software Repository to the user.

Chapter 2: Opsware Multimaster Mesh Administration

IN THIS CHAPTER

This chapter discusses the following topics:

- Opsware Multimaster Mesh Overview
- 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
- Resolving Model Repository Multimaster Component Conflicts

Opsware Multimaster Mesh Overview

This guide does not document how to set up the Opsware System to run in a multimaster mesh. For more information, see the *Opsware System 5.1 Deployment and Installation Guide* or consult your Opsware System Support Representative.

A multimaster mesh is a set of Opsware cores with synchronized Model Repositories. A mulitmaster 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 the Opsware System in a multimaster mesh has the following advantages:

- Redundancy – If a core in one facility becomes unavailable, the Opsware Command Center is still usable from other facilities. Users in other facilities can have their own Opsware Command Center. Also, it provides the ability to move out of a facility and keep the Opsware System 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 Opsware Command Center, a facility refers to the collection of servers that a single Opsware 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 Opsware Command Center in any facility. When a user updates data in a facility, the Model Repository for that facility is synchronized with the Model Repository databases located in all remote facilities. In the Opsware Command Center, a facility is identified by a facility name and a facility ID.

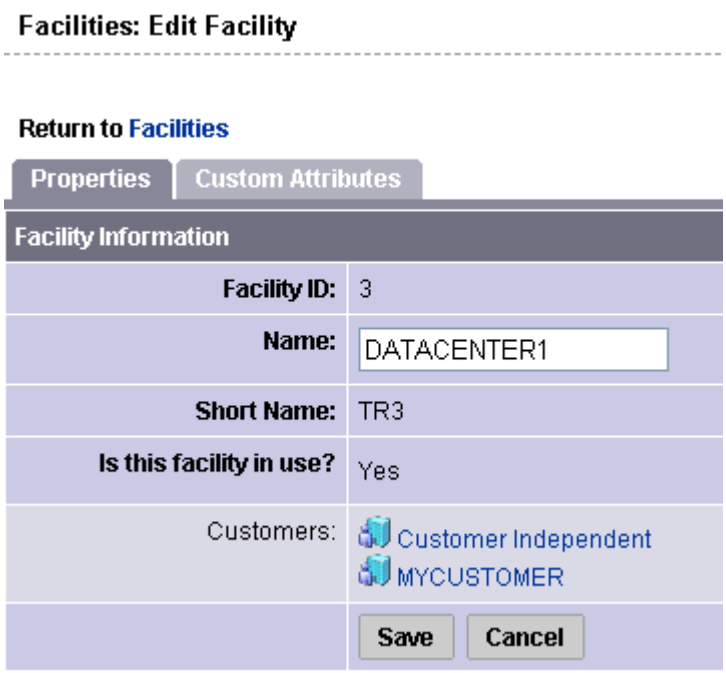
Updating Facility Information and Settings

Perform the following steps to update facility information and settings:

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

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 2-1 shows.

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



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

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

4 Click the Save button. The Opsware Command Center displays a message that confirms that the properties for that facility were updated.

5 Click 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 the Opsware System, 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 the New button.

Be careful when you update or remove existing attribute settings as it might affect or disrupt operation of the operational environment. Contact your Opsware System 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 the Opsware System and contains the following topics:

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

Multimaster Mesh Administration Overview

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 Opsware System components detect conflicts and propagate alerts; however, the multimaster components do not resolve conflicts. Opsware administrators use the multimaster tools in the Opsware Command Center to resolve the conflicts at the target databases.

In a configuration with multiple facilities, one facility, known as multimaster central, is designated as the primary facility. It is responsible for generating the transaction table automatically, although the other facilities can also do so upon demand. Multimaster central is automatically defined during installation. See “Designating the Multimaster Central Data Access Engine” on page 132 in Chapter 4 for more information

Model Repository Multimaster Component Conflicts

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

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

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

Data conflicts occur when the values of the objects in the local Model Repository do not match the values in a message from the Outbound Model Repository Multimaster Component or a database constraint is violated. When a conflict is flagged, the Opsware System 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 Opsware administrator must manually resolve the problem by using the Opsware Command Center. Resolving the conflict unlocks the rows. See “Best Practices for Resolving Database Conflicts” on page 58 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 Opsware Command Centers in different facilities. Conflicts occur when a user makes a change by using the Opsware Command Center in one facility and another user makes a change to the same object using the Opsware Command Center in another facility.

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

For example, this sequence of events occurs:

1 Alice removes Node A from a server in the Atlanta facility.

2. Bob removes Node A from the same server in the Boston facility.
3. The Opsware System 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. The Opsware System generates a Model Repository Multimaster Component conflict.
4. The Opsware System 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. The Opsware System 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 Opsware Command Centers, or between an Opsware Command Center and some command line utilities in a facility.

For example, this sequence of events occurs:

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

When the data arrives from Seattle, the Opsware System 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 the Opsware System is running from three or more facilities.

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

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

For example, this sequence of events occurs:

1. From a server in the Denver facility, Henry uses the OCLI to upload the package `henry.conf`.
2. The Opsware System 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 Opsware Command Center in Miami.
4. The Opsware System 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 the Opsware System in multiple facilities, try to keep the number of conflicts that can occur to a minimum. Educate users to consider the following factors when the Opsware System 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 Opsware System 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 72 in this chapter for more information.

  For additional assistance, consult your Opsware System Support Representative or see the *Opsware System 5.1 Deployment and Installation Guide* for information about network connectivity when running the Opsware System 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.

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

See the *Opsware System 5.1 Configuration Guide* for more information about User Groups and Opsware System 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 Opsware Command Center at all multiple facility installations.

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

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

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

See “Opsware System Diagnosis” on page 105 in Chapter 4 for more information.

Best Practices for Resolving Database Conflicts

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

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

Types of Conflicts

The following types of conflicts can occur:

- Identical data conflict – the Multimaster Tools show a conflicting transaction but the data is the same between facilities. The data is the same because users made the same change in different facilities.

- Simple transaction conflict – the row exists in all facilities, but some columns have different values or the row does *not* exist in some facilities (missing objects).
- Unique-key constraint conflict – the object does not exist in a facility and cannot be inserted there because inserting it would violate a unique-key constraint.
- Foreign-key constraint conflict – the row does not exist in some facilities and cannot be inserted because the data contains a foreign key to another object that also does not exist in that facility.
- Linked object conflict – a type of conflict encountered in rare cases. The Opsware System includes business logic that links specific related objects in the Opsware System, such as a custom attribute name and value, and a customer created in the Opsware Command Center UI (appears in lists) and the associated node for the customer in the node hierarchy. The Opsware System 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 Opsware System 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 depends 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 Opsware Command Center in the London facility, John creates Node A1 as a subordinate node of Node A.
2. From the Opsware Command Center in the San Francisco facility, Ann performs the same action. She creates Node A1 as a subordinate node of Node A.
3. Node names must be unique in each branch of the node hierarchy.
4. The Opsware System 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.

Resolving Model Repository Multimaster Component Conflicts

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

- Resolving Conflicts Overview
- Resolving a Conflict by Object
- Resolving a Conflict by Transaction

Resolving Conflicts Overview

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

Before you resolve conflicts, notify the subscribers of the email alert alias. Notifying these users helps to prevent other Opsware administrators from undoing or affecting each other's conflict resolution efforts.

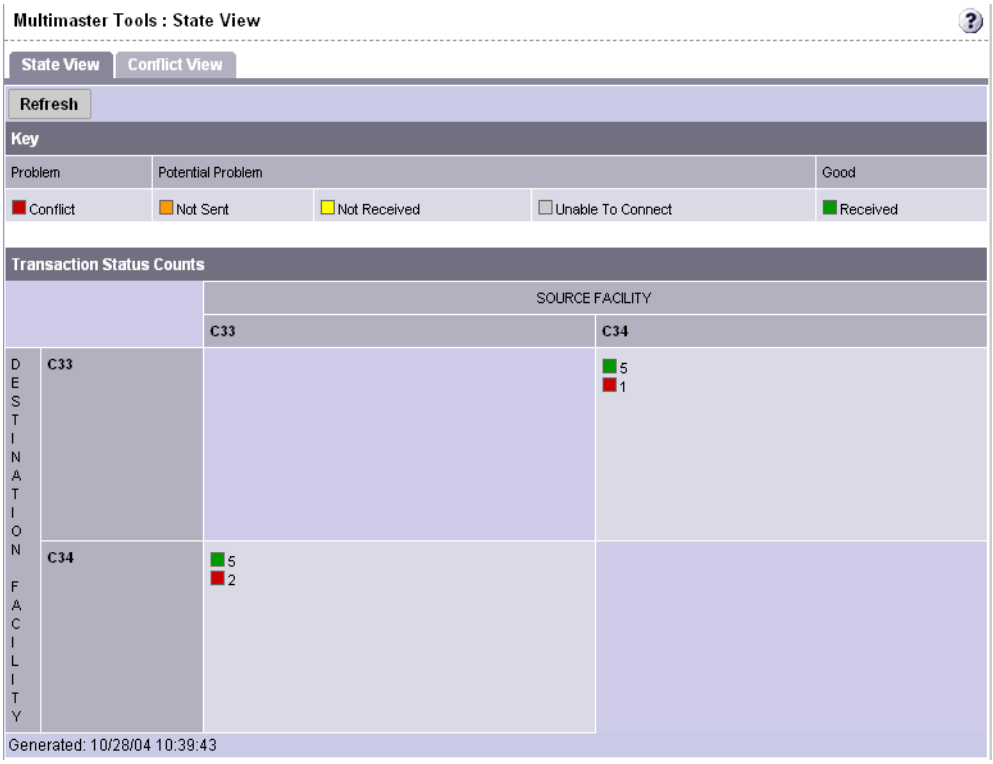
If you see a large volume of conflicts that you cannot resolve by using the Multimaster Tools, contact your Opsware System Support Representative for assistance synchronizing databases.

Resolving a Conflict by Object

Perform the following steps to resolve conflicting transactions by object:

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

Figure 2-2: Transaction Table That Shows Conflicts



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

- Green – the latest five transaction successfully sent
- Orange – all transactions that have not been published (sent to other facilities)
- Red – all conflicts

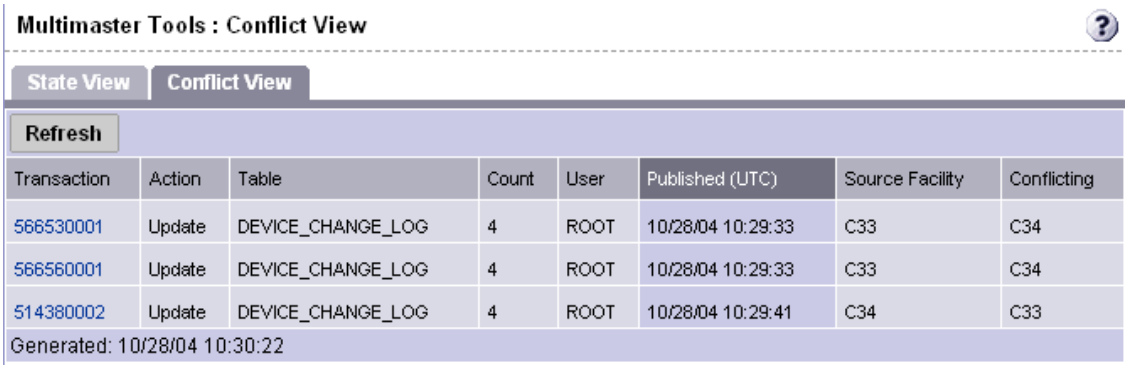
Each box is displayed in a color scheme to indicate the status and success of the transaction. A key that explains the significance of the colors, like the one shown in Figure 2-3, is listed at the top of the page.

Figure 2-3: Conflict Color Key

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

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

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



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

The page might show a conflict where the data is the same in both facilities but a conflict exists because the same change was made in both facilities. Even though the data is correct, the conflict still exists and must be resolved. See “Best Practices for Resolving Database Conflicts” on page 58 in this chapter for more information

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

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

Multimaster Tools: Transaction Differences | 566530001 from Source Facility C33

Return to Conflict View

Synchronize all objects from C34 Update

DeviceChangeLog 440001

DB Field	C34	C33
CHANGE_SUMMARY	SZXT3Aip fKk ZMv2cIBBe2pN2LdXSikB0GqKwdqG2 VbM7klQ R aWY s6T X olXPvRpRjqw HdRGgJPLg Bh CP7sSGgJfS1	h1V mflbM3lw IHQqj4i fd h nLB4 L044iK7Dg9qoYLK5wQkFnSgik J645XZYMjc wP FEMvhufpBlUqvf5fONOB 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
	Synchronize From	Synchronize From

DeviceChangeLog 450001

DB Field	C34	C33
CHANGE_SUMMARY	78vzxGYnqICbgqfjD StA1VU3LZkBSyY4 M NRJfPPRZyL WXVaINAr POOtheHMnLHMRA nX Ih J 1kQIK zMLr8I Yh YrFi	QuuM YPFNFH2cT 0wspWXvPZDGL9doTSvm9L8F z FfZ8yQPdW7Es 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
	Synchronize From	Synchronize From

DeviceChangeLog 460001

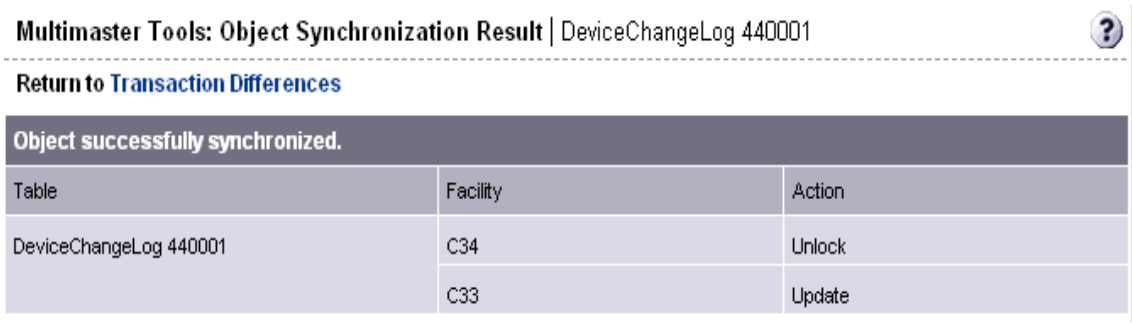
DB Field	C34	C33
CHANGE_SUMMARY	e M mwJWim6xPHM9nB0u mGOGX0 HPgQB443SzTrguhkx2P1t w A49w2 jE7QG99vuiznC rwC1ysjeB P sXsWrtZ8dx	OJjzp6C K FXuelN8Pcjg3KFe7 jukIaqTlVoTAEMdtiV0sA1Ew4ZPAwV c MXB0VXrEErDH yV w6Ryf 7l v pX
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	460001	460001

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

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

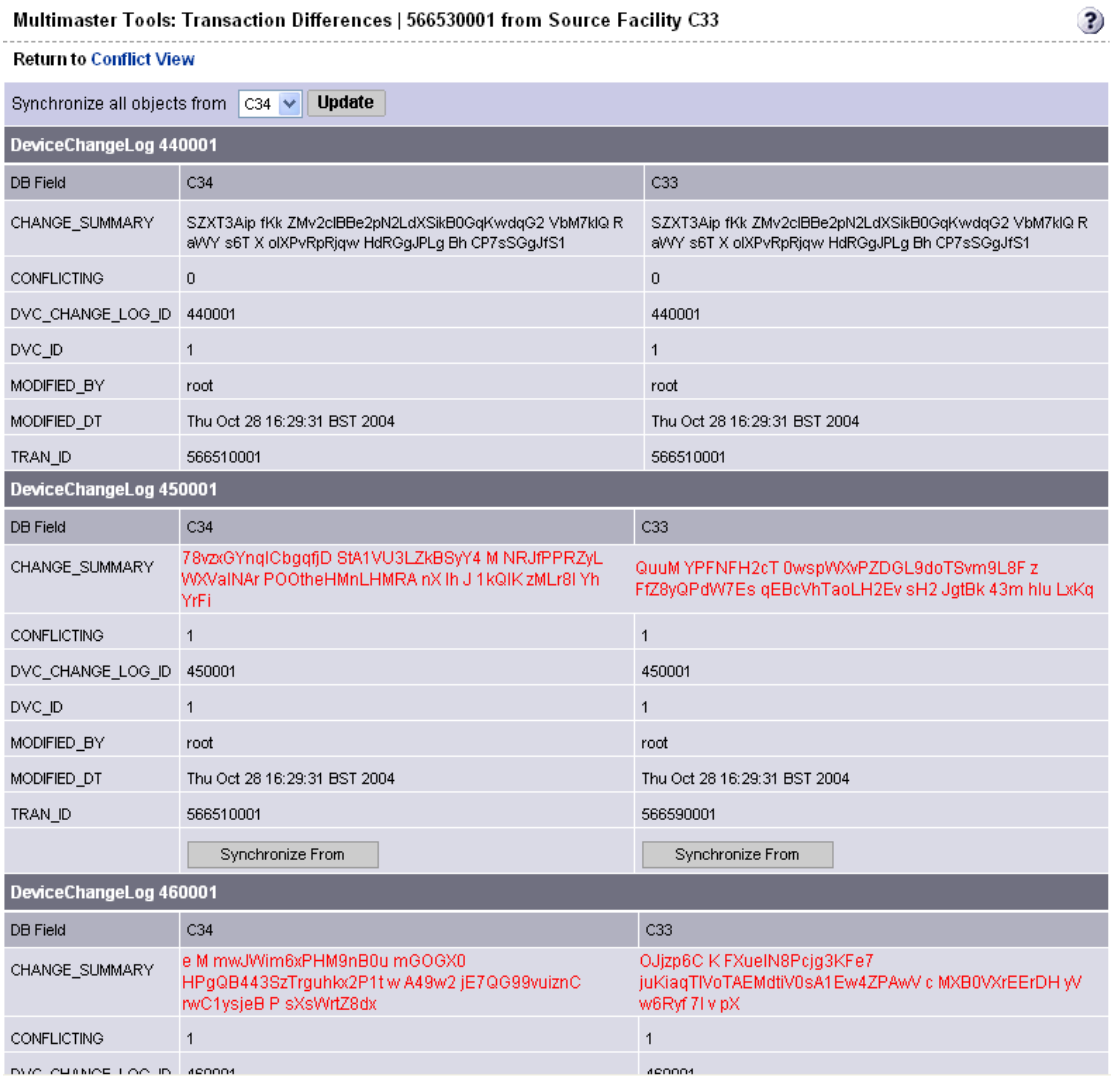
The Multimaster Tools: Object Synchronization Results page appears, displaying the results of the transaction synchronization, as shown in Figure 2-6.

Figure 2-6: Object Synchronization Result Page



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

Figure 2-7: Single Object Resolved



Multimaster Tools: Transaction Differences | 566530001 from Source Facility C33

Return to Conflict View

Synchronize all objects from C34 Update

DeviceChangeLog 440001

DB Field	C34	C33
CHANGE_SUMMARY	SZXT3Aip fKk ZMv2clBBe2pN2LdXSikB0GqKwdqG2 VbM7klQ R aWY s6T X olXPvRpRjqw HdRGgJPLg Bh CP7sSGgJfS1	SZXT3Aip fKk ZMv2clBBe2pN2LdXSikB0GqKwdqG2 VbM7klQ 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

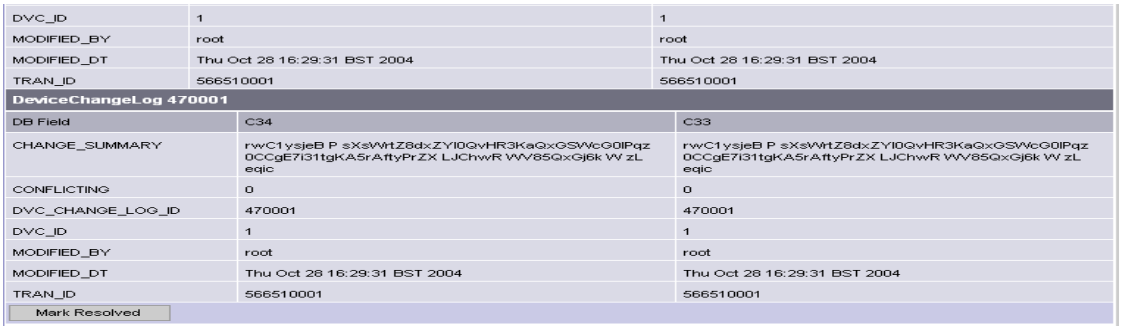
DB Field	C34	C33
CHANGE_SUMMARY	78vzxGYnqlCbgqfjD StA1VU3LZkBSyY4 M NRJfPPRZyL WXValNAr POOtheHMnLHMRA nX Ih J 1kQIK zMLr8l Yh YrFi	QuuM YPFNFH2cT 0wspWXvPZDGL9doTSvm9L8F z FfZ8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu LxKq
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	450001	450001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566590001
	Synchronize From	Synchronize From

DeviceChangeLog 460001

DB Field	C34	C33
CHANGE_SUMMARY	e M mwJWim6xPHM9nB0u mGOGX0 HPgQB443SzTrguhkx2P1t w A49w2 jE7QG99vuiznC rwC1ysjeB P sXsWrtZ8dx	OJjzp6C K FXueIN8Pcjg3KFe7 juKiaqTlVoTAEMdtiV0sA1Ew4ZPAwV c MXB0VXrEErDH yV w6Ryf 7l v pX
CONFLICTING	1	1

6 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, a Mark Resolved button appears at the bottom of the page, as Figure 2-8 shows.

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

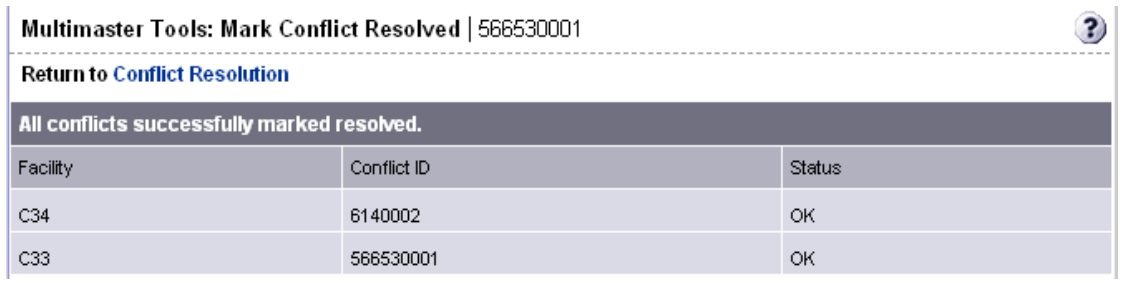


DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566510001
DeviceChangeLog 470001		
DB Field	C34	C33
CHANGE_SUMMARY	rwC1ysjeB P sXsWrtZ8dxZYI0QvHR3KaQxGSWcG0lPqz 0CCgE7I31tgKA5rAftyPrZX LJChwR WV85QxGj6k W zL eqic	rwC1ysjeB P sXsWrtZ8dxZYI0QvHR3KaQxGSWcG0lPqz 0CCgE7I31tgKA5rAftyPrZX LJChwR WV85QxGj6k W zL eqic
CONFLICTING	0	0
DVC_CHANGE_LOG_ID	470001	470001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566510001	566510001

Mark Resolved

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

Figure 2-9: Multimaster Tools Mark Conflict Resolved Page



Multimaster Tools: Mark Conflict Resolved | 566530001

Return to Conflict Resolution

All conflicts successfully marked resolved.

Facility	Conflict ID	Status
C34	6140002	OK
C33	566530001	OK

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

8 Click the link to return to the Conflict view.

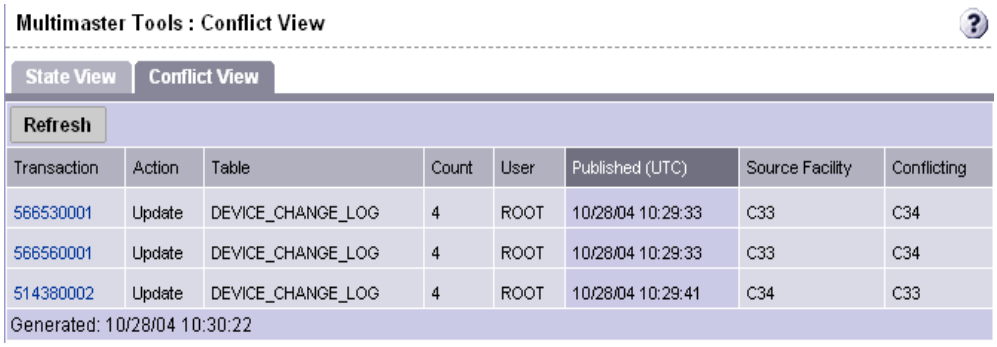
Resolving a Conflict by Transaction

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

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.

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

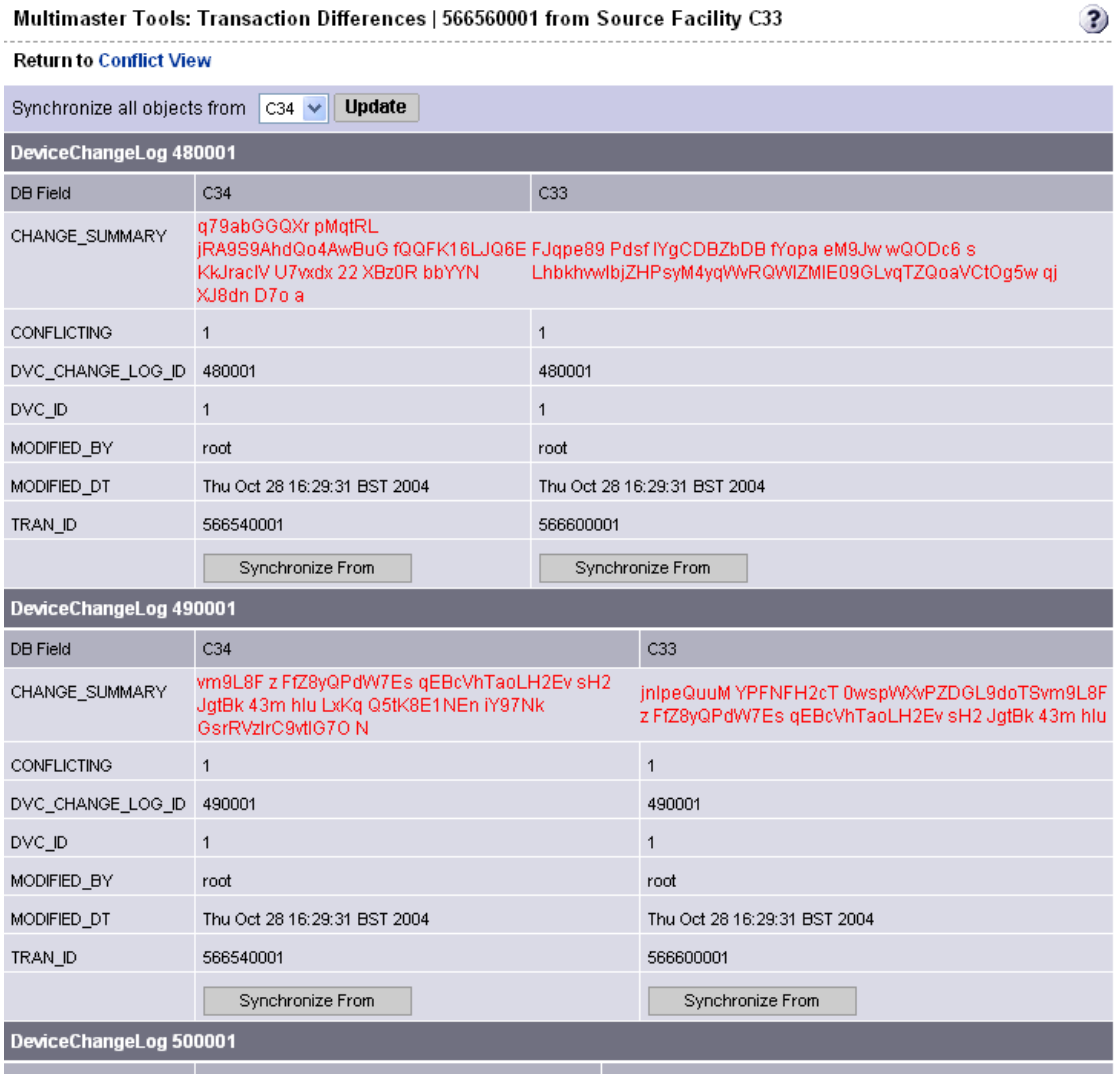
Figure 2-10: Transaction Differences Page That Lists all Transactions In Conflict



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

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

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



Multimaster Tools: Transaction Differences | 566560001 from Source Facility C33

Return to Conflict View

Synchronize all objects from C34 **Update**

DeviceChangeLog 480001

DB Field	C34	C33
CHANGE_SUMMARY	q79abGGQXr pMqtRL jRA9S9AhdQo4AwBuG fQQFK16LJQ6E KkJracIV U7vxdx 22 XBz0R bbYYN XJ8dn D7o a	FJqpe89 PdsfIYgCDBZbDB fYopa eM9Jw wQODc6 s LhbkhwwIbjZHPsyM4yqWwRQWlZMIE09GLvqTZQoaVCtOg5w qj
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
	Synchronize From	Synchronize From

DeviceChangeLog 490001

DB Field	C34	C33
CHANGE_SUMMARY	vm9L8F z FfZ8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu LxKq Q5tK8E1NEn iY97Nk GsrRVzlrC9vtlG7O N	jnlpeQuuM YPFNFH2cT 0wspWXvPZDGL9doTSvm9L8F z FfZ8yQPdW7Es qEBcVhTaoLH2Ev sH2 JgtBk 43m hlu
CONFLICTING	1	1
DVC_CHANGE_LOG_ID	490001	490001
DVC_ID	1	1
MODIFIED_BY	root	root
MODIFIED_DT	Thu Oct 28 16:29:31 BST 2004	Thu Oct 28 16:29:31 BST 2004
TRAN_ID	566540001	566600001
	Synchronize From	Synchronize From

DeviceChangeLog 500001

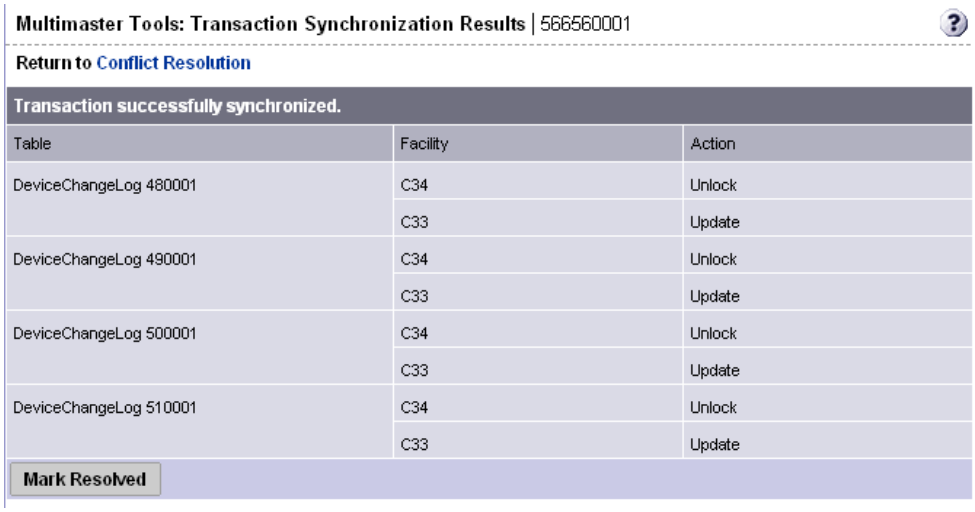
4 From the Synchronize all objects from drop-down list at the top of the page, select the facility to use as the correct source of data, as Figure 2-12 shows.

Figure 2-12: Resolving Conflicts by Transaction

See “Best Practices for Resolving Database Conflicts” on page 58 in this chapter for more information

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

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



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

6 Click the Mark Resolved button. The Multimaster Tools: Mark Conflict Resolved page appears. The page displays the results of marking a transaction resolved.

7 Click the link to return to the Conflict view. After it is marked resolved, the transaction disappears form the State and Conflicts views after the Opsware System refreshes the data in the Multimaster Tools.

Network Administration for Multimaster

Opsware Inc. 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.
- Opsware Command Centers cannot contact the multimaster central Data Access Engine.

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

Table 2-1: Performance Data for Multimaster Configurations

# FACILITIES	DURATION NETWORK OUTAGE	# MULTIMASTER CONFLICTS *
8 facilities (Opsware core installed in each facility)	12 hour outage (1 facility looses 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 Opsware Command Centers in other facilities increases the number of conflicts.		

Network connectivity issues include TIBCO or multicast routing problems.

Multimaster Alert Emails

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

This email address is configured when the Opsware System is installed in a facility. For assistance changing this email address, contact your Opsware System Support Representative or see the *Opsware System 5.1 Configuration Guide*.

The subject line of the alert email specifies:

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

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

See Table 2-2 for error messages.

Table 2-2: Multimaster Error Messages

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

Table 2-2: Multimaster Error Messages

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

Table 2-2: Multimaster Error Messages

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

Chapter 3: Opsware Satellite Administration

IN THIS CHAPTER

This chapter discusses the following topics:

- Opsware Satellite Overview
- Viewing Satellite Information
- Managing the Software Repository Cache
- Creation of Manual Updates

Opsware Satellite Overview

Opsware System 5.1 supports a new type of Opsware System installation – an Opsware Satellite. In an Opsware Satellite, a full Opsware core is not installed in the 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 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 System 5.1 Deployment and Installation Guide*.

In Figure 3-1, a Satellite is linked to a standalone core via the gateway and in Figure 3-2, two Satellites are linked to an Opsware core via the gateway. A Satellite can contain multiple gateways.

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

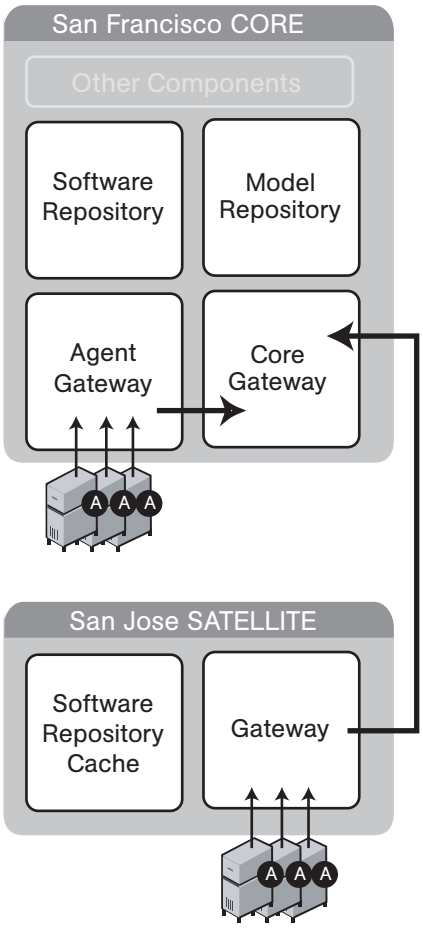
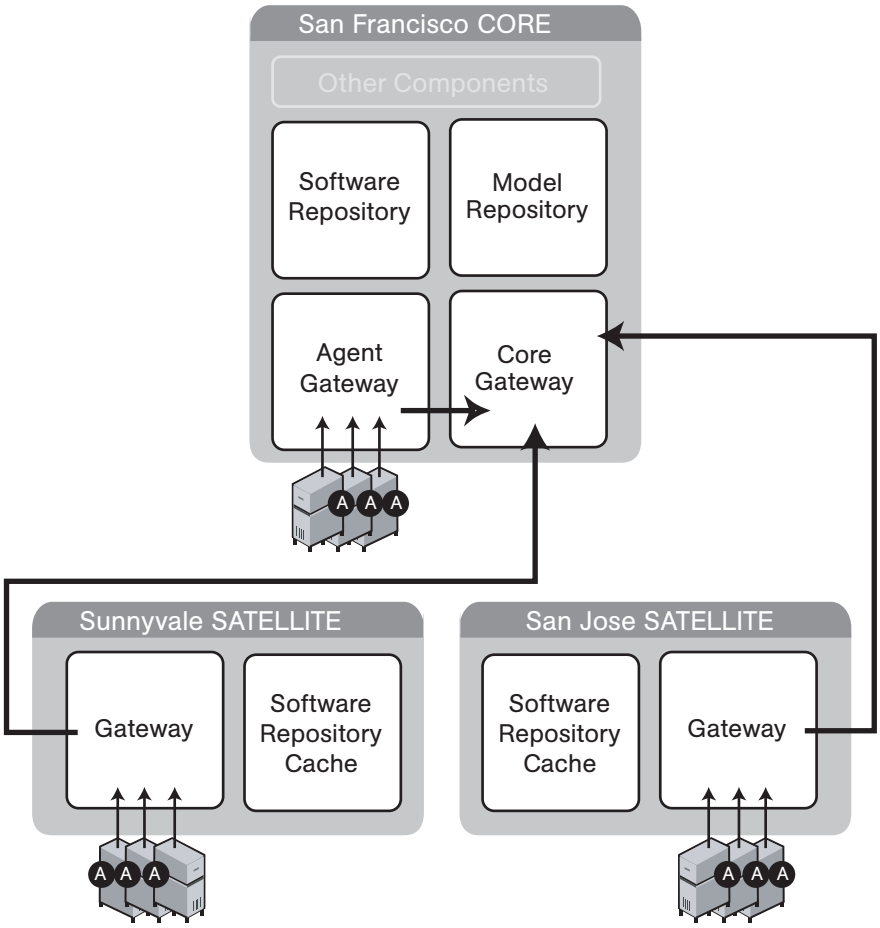


Figure 3-2: Standalone Core with Multiple Satellites



The Opsware Gateway

Connectivity with an Opsware core is achieved through an Opsware Gateway that resides in the same IP address space as the servers that it manages. This Opsware Gateway maintains a connection to the Opsware 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 Opsware Gateways.

Facilities and Realms

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

One or more Opsware Gateways service the managed servers contained within an Opsware realm. In the Opsware System, a realm is a routable IP address space, which is serviced by one or more gateways. All managed servers that connect to an Opsware 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.

Viewing Satellite Information

This section discusses the following topics:

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

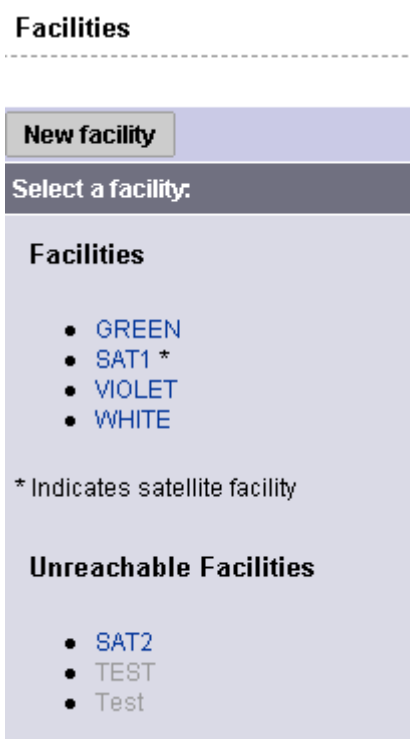
Permissions Required for Managing Satellites

To access the Manage Gateway feature, you must have the Manage Gateway permission. By default, this permission is included in the Opsware System Administrators group. To view facility information, you must have Read (or Read & Write) permission for the specific facility. See the *Opsware System 5.1 Configuration Guide* for information about user groups and Opsware permissions.

Viewing Facilities

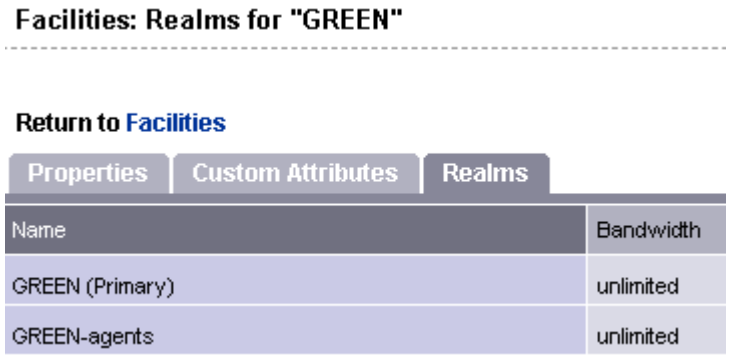
The Facilities page in the Opsware Command Center lists the core and Satellite facilities. In particular, the Facilities page displays Unreachable Facilities, as shown in Figure 3-3.

Figure 3-3: Facilities Channel



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

Figure 3-4: Realms in Facilities



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

Figure 3-5: Satellite Configuration Parameters

Enabling the Display of Realm Information

By default, the Opsware Command Center does not display realm information, which is needed by users who manage Gateways and Software Repository Caches.

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

1. Log in to the Opsware Command Center as a user that belongs the Administrators group and to a group that has the Configure Opsware permission.
2. From the navigation panel, select Administration ➤ System Configuration.
3. Select the Opsware Command Center link.
4. In the System Configuration page, for the name `owm.features.Realms.allow`, type the value `true`.
5. Click the **Save** button.

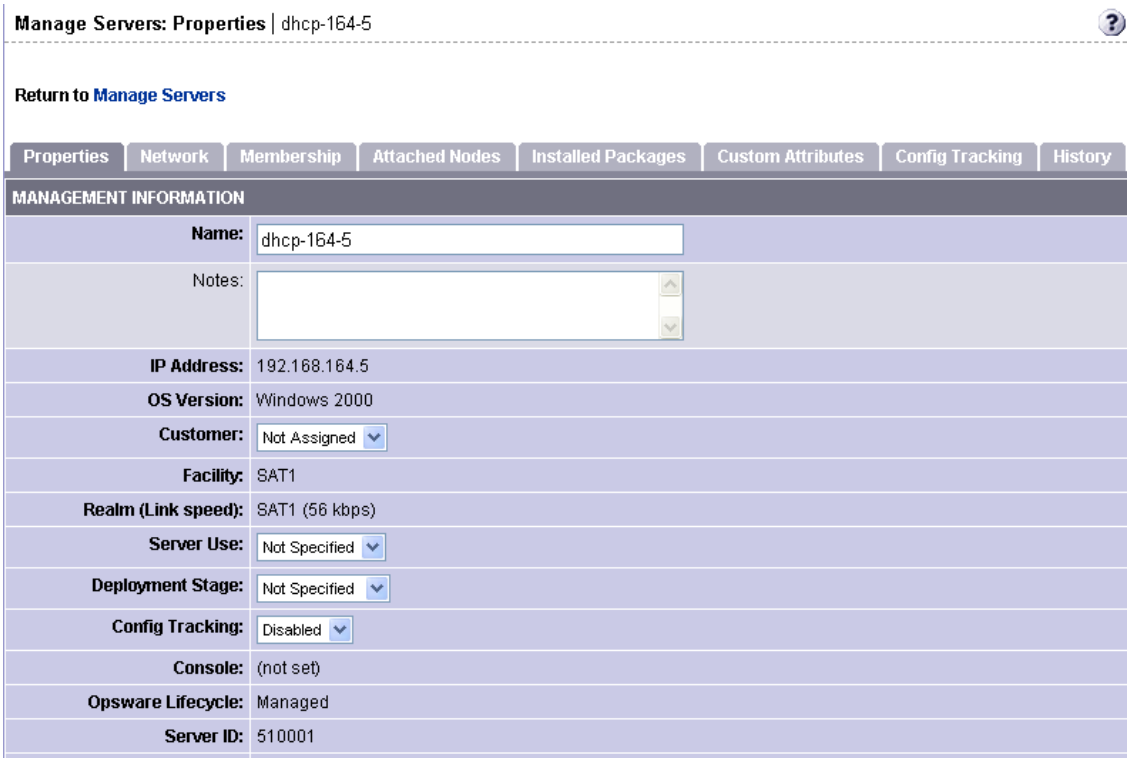
Viewing the Realm of a Managed Server

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

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

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

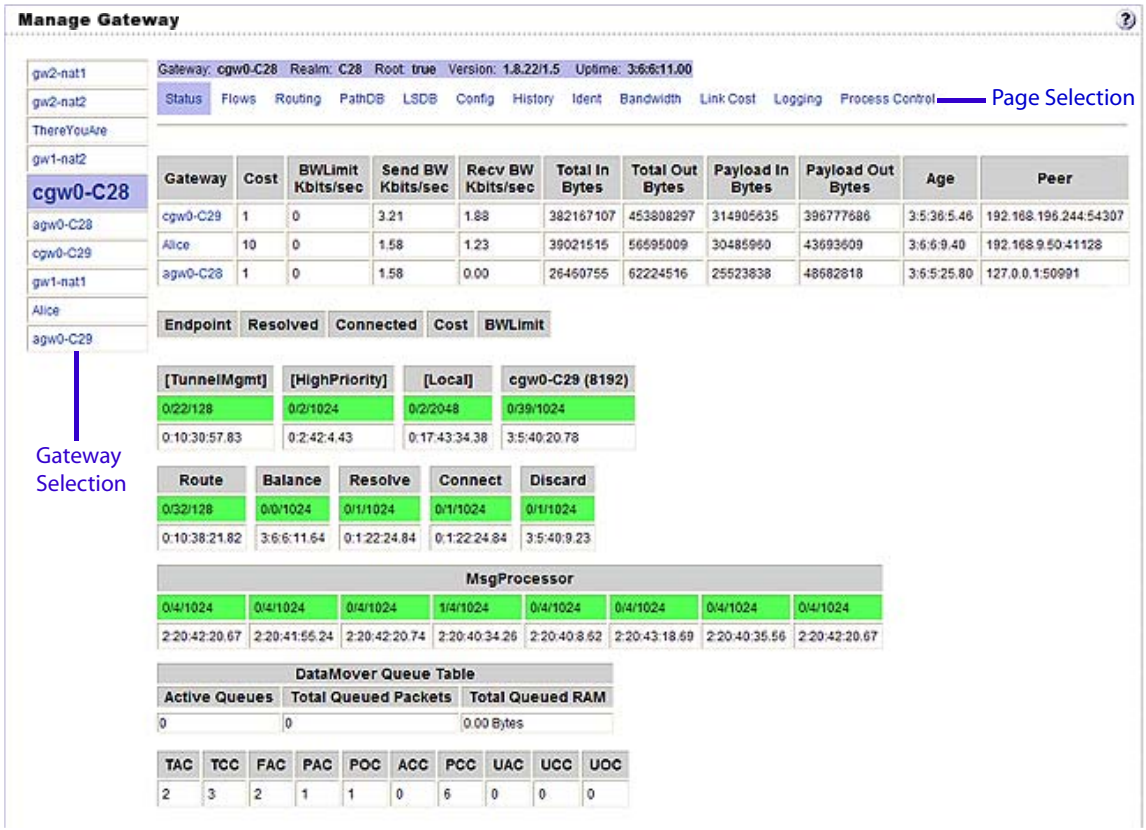
Figure 3-6: Server Properties Page Showing the Realm of a Managed Server



Viewing Gateway Information

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

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



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

To View Diagnostic and Debugging Information

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

   The **Status page** displays the following information for the gateway:

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

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

     You can use the timestamp indicating when the message high-water mark was last reached to troubleshoot gateway issues. The timestamp is displayed in the format days:hours:minutes: seconds.
3. To view the details and statistics for a tunnel between gateways, click the link for the gateway that terminates the tunnel, as Figure 3-8 shows.

Figure 3-8: Manage Gateway Feature – Status Page

   The page refreshes and displays the tunnel details and statistics.
4. To view the following pages containing diagnostic information, click the link for the page in the menu bar.

- **Flows page**: Displays information about all open connections for the selected gateway.
- **Routing page**: Displays the inter-gateway routing table. This table shows which tunnel will be used to reach another gateway in the mesh. The routing table is computed from the data in the path database.

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

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

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

  Above the properties values, the Properties Cache field appears. When you change the bandwidth or link cost for a connection between gateways, the updated value appears in this field if the update was successful.
- **History**: Displays historical information about the inbound (ingress) and outbound (egress) connections between hosts using the gateway mesh. For example, when host A in realm A connected to host B in realm B.

To Find the Source IP Address and Realm for a Connection

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

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

2. From the top bar (the page selector), click Ident. The page refreshes with an interface to the real-time connection identification database.
3. In the text field, enter the protocol and source port for an active connection (for example, TCP:25679).
4. Click **Lookup**.

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

To Change 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**.

To View 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.

To Restart or Stop 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.

Managing the Software Repository Cache

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 System performance in the Satellite will be poor unless special steps are taken, for example, installing a 1GB software package onto a server in a Satellite connected by a 56 kbps link will take a long time.

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

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

Availability of Packages on Software Repository Cache

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

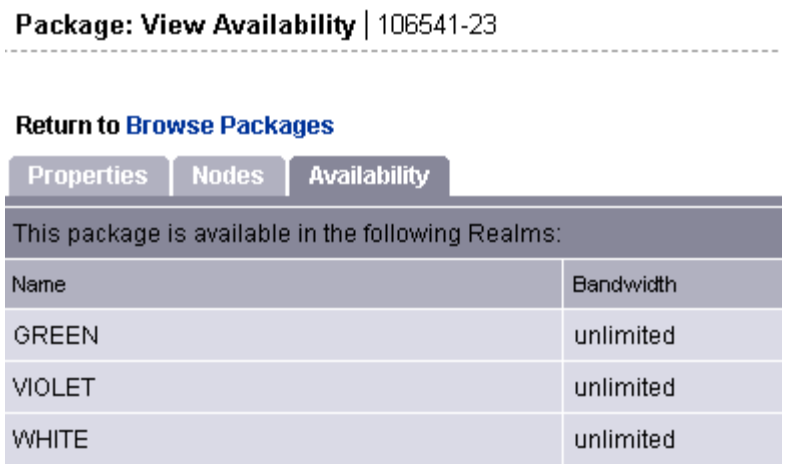
Instead, when the Opsware System is attempting to reconcile the software onto a managed server, the Opsware Command Center generates an Opsware error and displays a complete list of missing packages to help you identify the packages that need to be staged.

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

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

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

Figure 3-9: Package Availability



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

Ways to Distribute Packages to Satellites

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

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

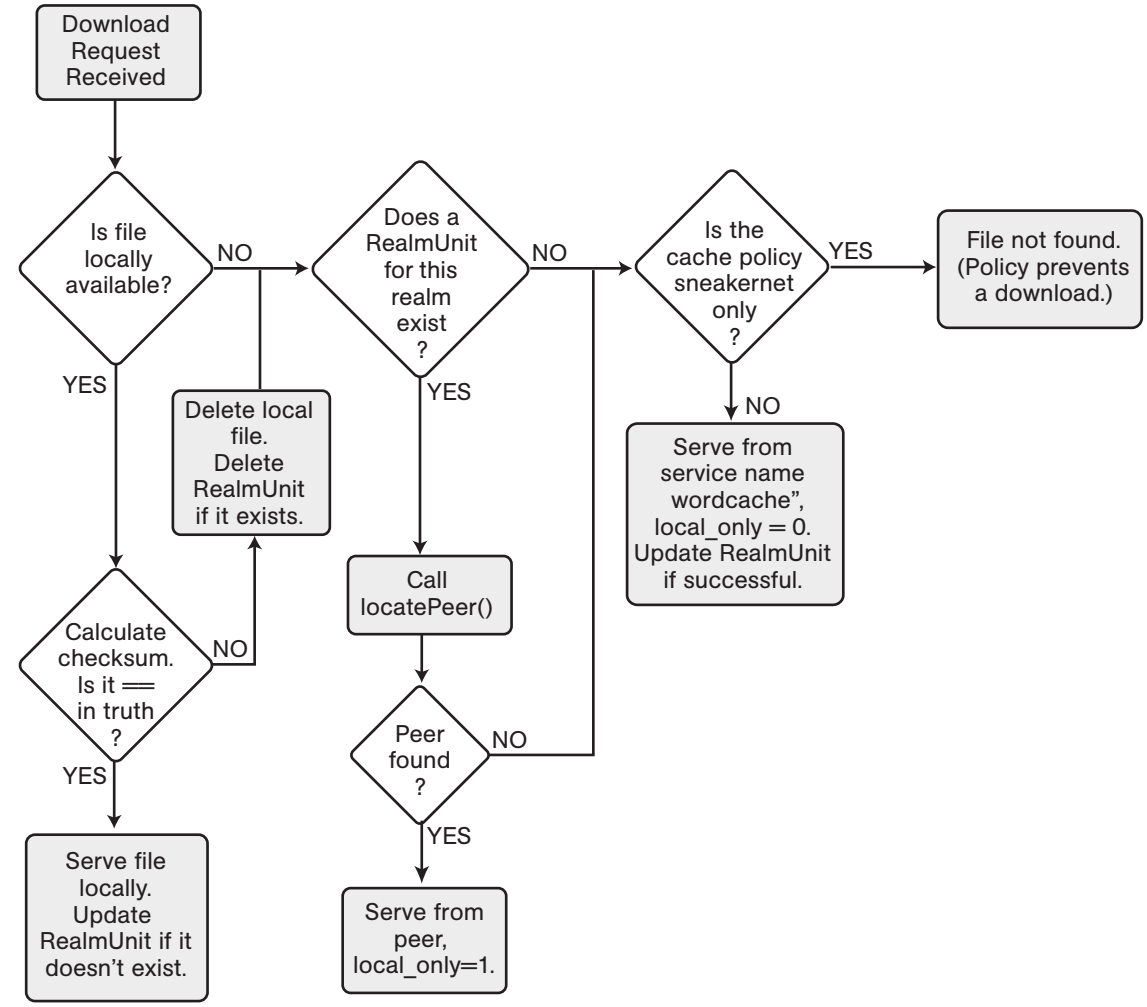
- **Manual Update:** software packages are staged to a Satellite's Software Repository Cache in advance of package installation so that performance will be about the same as if the managed server was in the same data center as the core.

It is always possible to stage a file on a Software Repository Cache regardless of the caching policy. See “Staging Files to a Software Repository Cache” on page 100 in this chapter for more information.

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

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

Figure 3-10: Software Repository Cache Update Logic

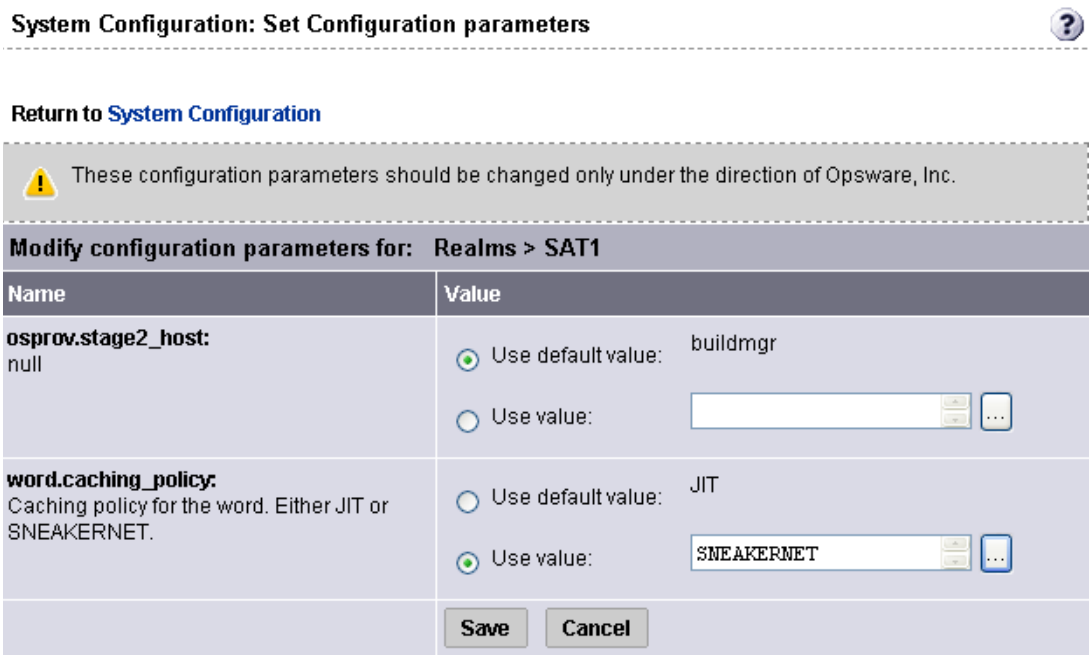


Setting the Update Policy

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

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

Figure 3-11: Software Repository Cache Configuration Parameters



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

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

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

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

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

Manual Updates

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

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

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

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

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

Hierarchical Software Repository Caches

When the Opsware System 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.

Opsware Inc. 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 Opsware DCML Exhange Tool (DET) to copy existing packages from an Opsware core. You can then save the exported file to CD or DVD to apply later to a Satellite Software Repository Cache.

The DET was formerly known as the Opsware Content Baseline Tool (CBT).

This section discusses the following topics:

- Creating a Manual Update Using the DCML Exhange 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 Exhange Tool (DET)

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

See the *Opsware System DCML Exchange Tool (DET) 2.0 Reference Guide* for more information about the DET. To create a Manual Update perform the following steps:

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

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

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

   ```
   opsware-ca.crt
   ```

spog.pkcs.8

to the following directory:

/usr/cbt/crypto

This is the directory where you installed the DET.

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

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

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

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



```
<directive rdf:resource="&filter;Descendants" />
</ApplicationFilter>
</rdf:RDF>
```

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

This filter will export from the Applications area of the Opsware Command Center. If you want to export packages from some other category of software in the Opsware Command Center, you need to create a different filter. See the *Opsware System DCML Exchange Tool (DET) 2.0 Reference Guide* for information.

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

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

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

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

   The packages are named `unitid_`*`nnnnnnn`*`.pkg`.

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

Applying a Manual Update to a Software Repository Cache

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

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

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

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

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

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

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

Staging Files to a Software Repository Cache

The Software Repository Cache allows an 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/lc/crypto/gateway/token.srv.
```

2 Log on to the server running Opsware Software Repository component.

3 Enter the following command to change directories:


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

4 To stage the files you want, run the utility `stage_pkg_in_realm` which has the following syntax:

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

Example Command to Run the Staging Utility

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

Password for admin: <password>

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

Microsoft Utility Uploads and Manual Updates

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

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

The Software Repository Cache allows a client to request that it obtain a file, but that it not actually send the file to the client. If the file is not already cached, the Software Repository Cache will obtain it from the parent Software Repository Cache if the caching policy allows it. To use this feature, the client includes the argument `checkonly=1` in the URL request for the file. See “Running the Staging Utility” on page 100 in this chapter for information about how to stage files.

Chapter 4: Opsware System Maintenance

IN THIS CHAPTER

This chapter provides an overview of:

- Possible Opsware System Problems
- Opsware System Diagnosis
- Logs for Opsware Components
- Restarting Opsware Components
- Opsware Software
- Overview of Mass Deletion of Backup Files
- Web Services Data Access Engine Configuration File

Possible Opsware System Problems

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

- Possible Opsware System Problems Overview
- Troubleshooting Opsware Components
- Contacting Opsware Inc. Support

Possible Opsware System Problems Overview

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

Many problems with the Code Deployment & Rollback (CDR) feature are caused by errors with the CDR configuration and setup. See the *Opsware System 5.1 Configuration Guide* for information about CDR configuration.

Troubleshooting Opsware Components

The following mechanisms for troubleshooting the Opsware System are available:

- Running the Opsware System Diagnosis tool (a tool for debugging common problems with Opsware components). See “Opsware System Diagnosis” on page 105 in this chapter for more information.
- Reviewing error logs for Opsware components. See “Logs for Opsware Components” on page 112 in this chapter for more information.
- Contacting Opsware Inc. Support.

Contacting Opsware Inc. Support

When you contact Opsware Inc. 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 System Diagnosis

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

- Opsware System Diagnosis Overview
- System Diagnosis Testing Process
- What the System Diagnosis Tests Check
- 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 System Diagnosis Overview

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 System Diagnosis tool.

System Diagnosis Testing Process

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

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

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

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

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

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

What the System Diagnosis Tests Check

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

The System Diagnosis tool tests the following components:

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

The System Diagnosis tool does not test the Build Manager.

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

Data Access Engine Tests

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

Stand-Alone Tests

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

Comprehensive Tests

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

Software Repository Tests

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

Stand-Alone Tests

None

Comprehensive Tests

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

Web Services Data Access Tests

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

Stand-Alone Tests

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

Comprehensive Tests

- Connect to the Web Services Data Access Engine

- Read a server record from the Model Repository and thereby checks connectivity to the Model Repository

Command Engine Tests

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

Stand-Alone Tests

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

Comprehensive Tests

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

Model Repository Multimaster Component Tests

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

Stand-Alone Tests

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

- Check the receiver health by examining the state of the Inbound Model Repository Multimaster Component

Comprehensive Tests

None

Running a System Diagnosis of Opsware Components

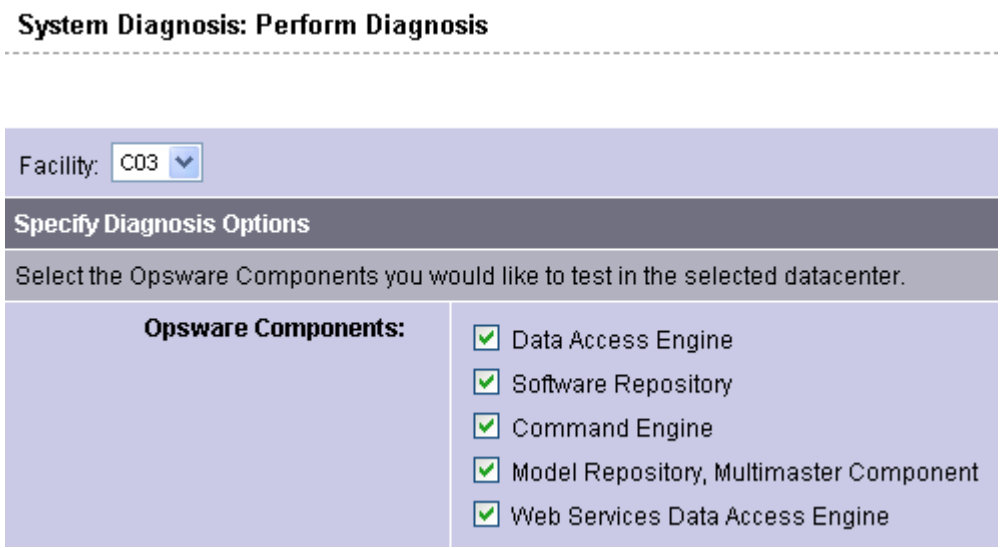
To access the System Diagnosis tool, you must have Opsware administrator privileges. See the *Opsware System 5.1 Configuration Guide* for more information about how to assign user privileges. The Opsware Command Center has access to all the Opsware Agents running on the Opsware component servers.

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

1. From the navigation panel, click Administration ➤ System Diagnosis. The System Diagnosis: Begin Diagnosis page appears.

2 Select the components that you want to test. By default, all components are selected (the Data Access Engine, the Software Repository, Command Engine, and Web Services Data Access Engine; in multiple core environments, there is also a selection for the Model Repository Multimaster Component). See Figure 4-1.

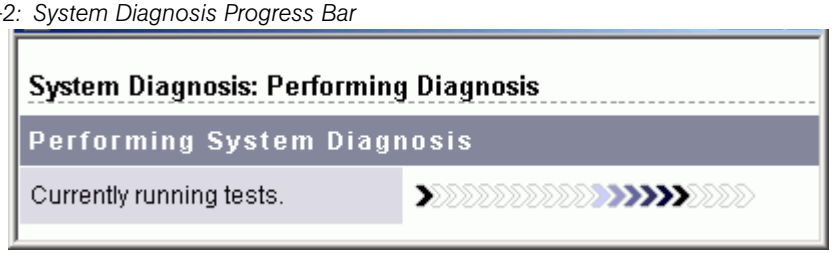
Figure 4-1: System Diagnosis Page That Shows Opsware Components Selected for Testing on the Indicated Facility



3 Click the Run Diagnosis button.

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

Figure 4-2: System Diagnosis Progress Bar



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

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

Logs for Opsware Components

This section provides information on Opsware component logs and contains the following topics:

- Opsware Component Logs Overview
- Boot Server Logs
- Build Manager Logs
- Command Engine Logs
- Data Access Engine Logs
- Media Server Logs
- Model Repository Logs
- Model Repository Multimaster Component Logs
- Opsware Agents Logs
- Opsware Command Center Logs
- Software Repository Logs
- Software Repository Replicator Logs
- Software Repository, Multimaster Component Logs
- Web Services Data Access Engine Logs
- Opsware Gateway Logs
- Global File System Server Logs

Opsware Component Logs Overview

Opsware components log events to a log-event handler and write human-readable summaries of log records, which are useful for troubleshooting.

By default, Opsware Inc. has configured the logging levels for the highest value (indicating higher priority) and the default log file size before rotation.

To increase or decrease the log levels for debugging or log file sizes in your organization's operational environment, contact your Opsware Support Representative for assistance.

You can access the logs for the Opsware components as files on each host running an Opsware component. The components use standard logging libraries for access and error logs, so the logs are output in a standard format. Use shell commands to view the error log files.

Refer to your own internal documentation where you have noted the Opsware component hostnames configured for your environment.

You can access the logs for the following components:

- Boot Server
- Build Manager
- Command Engine
- Data Access Engine
- Media Server
- Model Repository
- Model Repository, Multimaster Component
- Opsware Agents
- Opsware Command Center
- Software Repository
- Software Repository Replicator
- Software Repository, Multimaster Component
- Web Services Data Access Engine
- Opsware Gateway
- Global File System Server

Boot Server Logs

The Boot Server does not create its own logs. Installing the boot server component enables TFTP with INETD and the NFS server components. Additionally, the ISC DHCPD is installed. All these components log with syslog. Consult your vendor documentation for more information on these log files.

Build Manager Logs

Perform the following steps to access the Build Manager logs:

1. Access the host running the Build Manager by opening a terminal session.
2. Locate the following files:

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

Command Engine Logs

Perform the following steps to access the Command Engine logs:

1. Access the host running the Command Engine by opening a terminal session.
2. Locate the file:

   ```
   /var/lc/waybot/waybot.err
   ```

Data Access Engine Logs

Perform the following steps to access the Data Access Engine logs:

1. Access the host running the Data Access Engine by opening a terminal session.
2. Locate the file:

   ```
   /var/lc/spin/spin.err
   ```

Two Data Access Engine logs are in a multihost core.

Media Server Logs

Perform the following steps to access the Media Server logs:

1. Access the host running the Media Server by opening a terminal session.
2. Locate the following files relevant to Windows OS provisioning:

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

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

Solaris and Linux OS provisioning makes 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.

Model Repository Logs

The Model Repository is an Oracle database. The location of any logs that the database produces is specific to your installation.

Model Repository Multimaster Component Logs

Perform the following steps to access Model Repository Multimaster Component logs:

1. Access the host running the Model Repository Multimaster Component by opening a terminal session.
2. Locate the files:

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

*Where the files are `log.1, log.2, log.3`, and so forth. The size of the log file is configurable. Additional logs are created when the specified maximum file size is reached.

Opsware Agents Logs

Perform the following steps to access Opsware Agents logs:

1. Use a terminal session to access the host from which you need the Opsware Agent error log.
2. Locate the following file:

   `/var/lc/cogbot/cogbot.err` (UNIX-like systems)

   `%ProgramFiles%Common Files\loudcloud\cogbot\cogbot.err` (Windows systems)

Opsware Command Center Logs

The Opsware Command Center does not generate specific error logs. However, you can access the JBoss server log by performing the following steps:

1. Access the host running Opsware Command Center by opening a terminal session.
2. Locate the following file:

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

Software Repository Logs

Perform the following steps to access the Software Repository logs:

1. Access the host running the Software Repository by opening a terminal session.
2. Locate the following files:

   ```
   /var/lc/mm_wordbot/wordbot.err*
   ```

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

Software Repository Replicator Logs

Perform the following steps to access the Software Repository Replicator logs:

1. Access the host running the Software Repository, Multimaster Component by opening a terminal session.
2. Locate the following files:

   ```
   /var/lc/replicator/replicator.err
   ```

   ```
   /var/lc/replicator/daemonbot.out
   ```

   ```
   /var/lc/replicator/replicator.log.*
   ```

   ```
   /var/lc/replicator/1
   ```

*Where the files are `log.1, log.2, log.3`, and so forth. The size of the log file is configurable. Additional logs are created when the specified maximum file size is reached.

Software Repository, Multimaster Component Logs

Perform the following steps to access the Software Repository, Multimaster Component logs:

1. Access the host running the Software Repository, Multimaster Component by opening a terminal session.
2. Locate the following files:

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

*Where the files are `log.1, log.2, log.3`, and so forth. The size of the log file is configurable. Additional logs are created when the specified maximum file size is reached.

Web Services Data Access Engine Logs

The Web Services Data Access Engine logs any errors or exceptions in its standard out log. JBoss-specific messages are generated in the JBoss server log.

1. Access the host running the Web Services Engine by opening a terminal session.
2. Locate the following files:

   `/var/lc/twist/stdout.log`

   `/var/lc/twist/twist.log`

   `/var/lc/twist/access.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.

The `twist.log` file contains JBoss-specific error or informational messages.

The `access.log` file contains access information in common log format.

Opsware Gateway Logs

Perform the following steps to access the Opsware Gateway logs:

1. Access the host running the Opsware Gateway component by opening a terminal session.
2. Locate the following files:

   `/var/opt/OPSWgw/<name>`

   Where the `<name>` is the name of the Opsware Gateway instance.

Global File System Server Logs

Perform the following steps to access the Global File System Server logs:

1. Access the host running the Global File System Server component by opening a terminal session.
2. Locate the following files:

   `/var/opt/OPSWhub`

   `/var/opt/OPSWogfs`

Restarting Opsware Components

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

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

Restarting Opsware Components Overview

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

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

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

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

You must restart the following components in the following order:

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

   Stop the Opsware components that are running on the same host. If you leave the components running, they spawn a new `rvd` (not `rvrd`) process.

   Stop the `rvrd` process: `/etc/init.d/rvrdscript stop`

   Start the `rvrd` process.

   Start the Opsware components that are running on the same host.
6. In a multiple facility environment, if you need to restart the Web Services Data Access Engine, in addition to the Data Access Engine, and the Model Repository, you must also be sure that the Model Repository Multimaster Component is running first.

Restarting the Boot Server

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

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

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

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

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

Restarting the Build Manager

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

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

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

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

Restarting the Command Engine

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

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

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

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

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

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

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

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

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

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

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

Restarting the Data Access Engine

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

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

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

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

Restarting the Media Server

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

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

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

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

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

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

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

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

Restarting the Model Repository

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

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

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

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

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

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

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

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

Restarting the Model Repository Multimaster Component

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

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

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

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

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

Restarting an Opsware Agent

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

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


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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

On Windows, enter the following commands:

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

Restarting the Opsware Command Center

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

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

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

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

Restarting the Software Repository

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

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

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

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

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

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

Restarting the Software Repository, Multimaster Component

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

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

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

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

Restarting the Web Services Data Access Engine

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

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

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

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


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

Restarting the Opsware Gateway

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

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

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

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

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

Restarting the Global File System Server

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

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

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

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

Opsware Software

The Opsware Software function is populated during the Opsware System installation.

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

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

Table 4-1: Opsware Internal and External Component Names

INTERNAL NAME	EXTERNAL NAME
Agent	Opsware Agent

Table 4-1: Opsware Internal and External Component Names

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

Some of the functionality available in the Server Management area of the system is also available to be applied to the servers that appear on the Members tab. Take care in applying changes to the core servers. In particular, do not assign or unassign servers to these nodes or install or uninstall software or change networking unless directed to do so during the installation process by the *Opsware System 5.1 Deployment and Installation Guide*.

To view the servers on which each component is installed, click the component's hyperlinked name, then click 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 click the tab.

The *Opsware System 5.1 Deployment and Installation Guide* has the following two topics that deal with the Opsware Software function:

- "Reassigning the Data Access Engine to a Secondary Role"
- "Designating the Multimaster Central Data Access Engine"

Overview of Mass Deletion of Backup Files

The Opsware System 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 Opsware Command Center can be sluggish, and the information that displays might be cluttered with out-of-date configuration tracking data.

When the backup deletion script is run, it deletes all backed up files with the exception that it always keeps one copy of the latest version of every file ever backed up. If you want to delete those files, use the process for deleting backups individually or a few at a time that is covered in the *Opsware System 5.1 User's Guide*.

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

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

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

Syntax for the Command

`backup_delete.pyc` [*options*]

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

Using the Mass Deletion Script to Delete Backup Files

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

1. Log in as root to the server where the Data Access Engine is installed.
2. Make sure that `/lc/blackshadow` is in your PYTHONPATH environment variable.
3. Create a file that contains the arguments and values that you want the Opsware System to use with the mass deletion script. See Table 4-2 on page 128, 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
emailsuccess: 1
```

Table 4-2: Configuration File Options

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

Table 4-2: Configuration File Options

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

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

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

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

The crontab entry *must* be all on one line.

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

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

Designations for Multiple Data Access Engines

This section discusses the following topics:

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

Overview of Designations for Multiple Data Access Engines

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

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

- **Multimaster Central Data Access Engine** - An Opsware multimaster mesh of cores has only one multimaster central Data Access Engine. Although any of the cores may have multiple Data Access Engines, only one engine in the mutilmaster mesh can be the central engine.

Reassigning the Data Access Engine to a Secondary Role

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

1. Log in to the Opsware Command Center as a user that belongs to the Opsware System Administrators group.

   The Opsware Command Center should be installed and listening. The Opsware Command Center home page appears.

2. Click Administration ➤ Opsware Software from the navigation panel. The Opsware Software page appears.
3. Click the spin link. The Opsware Software | spin page appears.
4. Click 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, choose Re-Assign Node.
7. Select the radio button for the Service Levels | Opsware | spin node.
8. Click the Select button.
9. Navigate the node hierarchy by clicking the following nodes:
   - Opsware
   - spin
   - Secondary
10. Click the Re-Assign button.
11. In a terminal window, log in as root to the server running the additional Data Access Engine and enter the following command to restart the Data Access Engine:

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

Designating the Multimaster Central Data Access Engine

The Opsware Installer automatically assigns the multimaster central Data Access Engine. You need to perform the following procedure only if you need to change the default designation. For example, you might need to change the designation based on bandwidth requirements in a facility.

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

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

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

Web Services Data Access Engine Configuration File

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

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

To change a property defined in the configuration file:

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

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

The following table lists the properties of the configuration file. Several of these properties are related to the cache (sliding window) of server events. The Opsware System maintains a sliding window (with a default size of two hours) of events describing changes to `Server` objects. This window makes it possible for software developers to update a client-side cache of `Server` objects without having to retrieve all of the objects. For more information, see the `getEventList` operation of the Server Web Service in the *Opsware System Web Services API 3.0 Guide*.

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

PROPERTY	DEFAULT	DESCRIPTION
`twist.webservices.debug.level`	`1`	An integer value that sets the debug level for the Opsware Web Services API on the server side. Allowed values: 0 - basic info 1 - more detailed information 2 - stack trace 3 - for printing the server event cache entries whenever there is an item added to the cache.

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

PROPERTY	DEFAULT	DESCRIPTION
`twist.webservices.locale.country`	`US`	The country Internationalization parameter for the Localizer utility. Currently only the `US` code is supported.
`twist.webservices.locale.language`	`en`	Sets the language Internationalization parameter for the Localizer utility. Currently only the `en` code is supported.
`twist.webservices.caching.windowsize`	`120`	In minutes, the size of the sliding window maintaining the server event cache.
`twist.webservices.caching.windowslide`	`15`	In minutes, the sliding scope for the window maintaining the server event cache.
`twist.webservices.caching.safetybuffer`	`5`	In minutes, the safety buffer for the sliding window maintaining the server event cache.
`twist.webservices.caching.minwindowsize`	`30`	In minutes, the minimum size of the sliding window that maintains the server event cache.
`twist.webservices.caching.maxwindowsize`	`240`	In minutes, the maximum size of the sliding window that maintains the server event cache.

Index

J

L

M

N

O

P

R

S

T

U

V

W

