



Opsware[®] SAS 6.5 Policy Setter's Guide

Corporate Headquarters

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

Opware SAS Version 6.5.1

Copyright © 2000-2007 Opware Inc. All Rights Reserved.

Opware Inc. Unpublished Confidential Information.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

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

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

Additional proprietary information about third party and open source materials can be found at <http://www.opware.com/support/sas651tpos.pdf>.

Table Of Contents

Preface	15
<hr/>	
Overview of this Guide	15
Contents of this Guide	15
Conventions in this Guide	16
Icons in this Guide	17
Guides in the Documentation Set and Associated Users	17
Opsware, Inc. Contact Information	18
Chapter 1: Server Management Configuration	21
<hr/>	
Ways to Configure Server Management	21
Supported Operating Systems for Managed Servers	21
Customer Accounts in Opsware SAS	24
Associated Servers with Customers	26
Customer Account Administration	28
Creating a Customer	28
Updating Customer Information and Settings	30
Setting Custom Attributes for Customers	32
Restrictions for Deleting Customers	33
Deleting a Customer	34
Server Attributes	35

Creating Server Use Values	36
Editing Server Use Values.	37
Deleting Server Use Categories	38
Creating Deployment Stage Values	38
Editing Deployment Stage Values.	40
Deleting Deployment Stage Values	40
IP Range Groups and IP Ranges	41
Overview of IP Range Groups and IP Ranges.	41
Creating an IP Range Group	43
Creating an IP Range.	44
Changing Address Ranges on IP Ranges.	45
Increasing and Decreasing the Prefix Length.	46
Changing the Status of an IP Address in an IP Range	47
Chapter 2: Software Management Setup	51
Overview of Software Management	51
Policy-Based Software Management	51
Software Management Features	53
Software Management Process	54
Software Management Setup Tasks	55
Library	56
Overview of Software Policies	57

Software Policy Inclusion	58
Setting Custom Attributes For Software Policies.	60
Including ISM Controls in Software Policies	61
Software Policies for Patch Installation	61
Creating a Software Policy	62
Ways to Open a Software Policy.	64
Setting Software Policy Properties	66
Adding a Package to a Software Policy	68
Adding a Patch to a Software Policy	70
Adding an RPM Package to a Software Policy	71
Adding an Application Configuration to a Software Policy.	73
Adding a Software Policy to a Software Policy.	74
Specifying the Installation Order in a Software Policy	76
Setting Installation and Update Options for an RPM Package	77
Removing a Package from a Software Policy	79
Removing a Patch from a Software Policy	80
Removing an RPM Package From a Software Policy	80
Removing an Application Configuration from a Software Policy	81
Removing a Software Policy from a Software Policy.	82
Adding Custom Attributes to a Software Policy.	82
Editing Custom Attributes in a Software Policy	83
Deleting Custom Attributes from a Software Policy	83
Adding Custom Attributes to Servers.	84
Duplicating Zip Packages	84
Editing the ZIP Installation Directory	85
Viewing Servers Attached to a Software Policy	86
Viewing All the Software Policies Associated with a Software Policy	87

Viewing the History of a Software Policy87
Locating Software Policies in Folders87
Folders87
Folders and Permissions89
Creating a Folder90
Setting Folder Properties91
Deleting a Folder93
Overview of Package Management93

AIX Packages	95
HP-UX Packages.....	96
Linux Packages.....	100
Solaris Packages.....	101
Windows Packages	103
ZIP Packages.....	104
Windows Performance for Uploading Packages	106
Character Encoding for Package Metadata and Scripts	107
Importing a Package	109
Exporting a Package	111
Ways to Open a Package	111
Viewing Package Properties.....	112
Editing Package Properties	114
Viewing Package Contents.....	116
Viewing Servers Associated with a Package.....	117
Viewing All Software Policies Associated with a Package	117
Deleting a Package	117
Renaming a Package.....	118
Locating Packages in Folders	118
RPM Deployment	119

Install and Update of RPM Packages Using a Software Policy	120
Automatically Updating RPM Packages in a Software Policy	121
Upgrade Option For an RPM Package	122
Uninstalling RPM Packages	125
Software Policy Compliance for RPM Packages	126
Automatically Importing Red Hat Network Errata	127
Viewing Errata Based and Channel Based Software Policies in the SAS Client.	128
Chapter 3: Operating System Provisioning Setup	133
OS Provisioning Setup	134
Overview.	134
OS Provisioning Setup Tasks	135
Setting Up Sun Solaris OS Provisioning.	135
Setting Up Linux or VMware ESX OS Provisioning	136
Setting Up Microsoft Windows OS Provisioning.	138
OS Media Management	139
Overview.	140
Setting Up the Media Server	141
Import Media Tool Syntax and Options	143
Creating an MRL with the Import Media Tool.	146
Editing an MRL	147
Deleting an MRL.	148
OS Installation Profiles	148

Overview.....	149
Specifying Software in OS Installation Profiles.....	150
Configuration Files.....	151
Sun Solaris Profiles.....	151
Red Hat Linux Configuration Files.....	151
VMware ESX Configuration Files.....	152
SUSE Linux Configuration Files.....	152
Microsoft Windows Response Files.....	153
Using OS Installation Profiles.....	155
Defining an OS Installation Profile – Unix.....	155
Hardware Signature Files for Windows.....	158
Defining an OS Installation Profile – Windows.....	159
Modifying Existing OS Installation Profiles.....	163
Changing the OS Installation Profile Properties.....	164
Modifying How an OS Is Installed on a Server – Unix.....	164
Modifying How an OS Is Installed on a Server - Windows.....	165
Modifying the OS Installation Profile Packages.....	167
Viewing Change History for an OS Installation Profile.....	169
Deleting an OS Installation Profile.....	170
Build Customization Scripts.....	171

Using Build Customization Scripts	171
Before Creating a Sun Solaris Build Customization Script.....	172
Solaris Provisioning from a Boot Server on a Redhat Linux Server ...	175
Solaris Build Customization Script	176
Requirements for Solaris Build Customization Scripts	176
Sample Solaris Build Customization Script	177
Linux Build Process	179
Linux Build Customization Scripts.....	181
Requirements for Linux Build Customization Scripts	181
VMware ESX Build Process	182
VMware ESX Build Customization Scripts.....	182
Windows Build Process (DOS Boot Image).....	183
Windows Build Process (WinPE Boot Image).....	184
Windows Build Customization Scripts (DOS Boot Image).....	186
Default Custom Attribute Values	190
Custom Attributes for Sun Solaris.....	191
Custom Attributes for Linux or VMware ESX	192
Custom Attributes for Microsoft Windows	193
Adding Custom Attributes to OS Installation Profile (SAS Web Client)	195
Adding Custom Attributes to OS Installation Profile (SAS Client)	196
Virtualization Support – VMware ESX (and Solaris 10)	197
Hardware Support in OS Provisioning	197

Overview of Hardware Support in OS Provisioning	198
PXE Images for Windows and Linux or VMware ESX	198
Windows and Linux or VMware ESX Boot Images.	199
NIC Support in Windows Boot Images.	199
Adding NIC Support to a Windows Boot Image.	201
Sample Mapfile	202
Sample Mapfile for an Intel 8255x-based PCI Ethernet Adapter	202
Prerequisites for Creating Windows Boot Images.	203
Creating a Windows Boot Image.	203
Updating PXE Image for Windows	204
Adding Hardware Support to a Linux or VMware ESX Build Image	205
Creating a Linux or VMware ESX Boot Image	206
Example: Usage of the OPSWlinuxbootiso Utility	207

Chapter 4: Code Deployment Setup **209**

Opsware Code Deployment Process	209
Deploying Code	209
Uploading Code and Content to Staging.	211
CDR Operations and Directories.	212
CDR Features	213
CDR Permissions	214
Accessing CDR	215
Code Deployment & Rollback Setup.	216

Prerequisites for Code Deployment & Rollback.....	217
Code Deployment Configuration Checklist.....	218
Planning and Defining a CDR Configuration.....	218
Configuring a Site to Use CDR for Code and Content Updates.....	219
Code and Content Deployment Requirements.....	220
Overview of CDR Configuration Planning.....	221
Preparing Host Machines.....	225
Creating or Verifying Directories on Hosts.....	225
Initial Content in Directories.....	226
Access Control for CDR.....	226
Defining CDR Services, Synchronizations, and Sequences.....	229
CDR Service Management.....	229
Defining a Service.....	230
Pre-Synchronization and Post-Synchronization Scripts.....	235
Modifying a Service.....	235
Deleting a Service.....	236
CDR Synchronization Management.....	237
Defining a Synchronization.....	237
Modifying a Synchronization.....	240
Deleting a Synchronization.....	240
CDR Sequence Management.....	241
Defining a Sequence.....	242
Modifying a Sequence.....	245
Deleting Sequences.....	245
Verifying and Troubleshooting CDR Configuration.....	246
Chapter 5: Visual Packager	247

Overview of Visual Packager247
Packaging Server Setup248
Configuring Options for a Packaging Server	251
Overview of Packages252
Packaging Process254
Ways to Create a Package255
Creating a Package from a Managed Server	255
Creating a Package from Audit Results	256
Creating a Package from a Snapshot	257
Creating a Package257
Adding New Package Content261
Specifying Options for New Package Content264
Viewing Package Details267
Appendix A: CML Tutorial and Reference	269
About the CML Tutorial269
CML Fundamentals270
What is an Application Configuration Template?	270
What is an Application Configuration?	270
What is CML?	271
About the CML Parser	271
Anatomy of a CML Tag	271
CML Tags You Should Know	273
Creating a CML Template275
Materials Needed for the Tutorial	275
Completed Template Sample	275
Completed url_scan_ini.tpl Template297

Using DTD Tags in CML	300
DTD Tags Example.....	300
Sequence Aggregation	301
Sequence Replace	303
Sequence Append.....	303
Sequence Prepend.....	305
CML Grammar	306
CML Options	309

Preface

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

Overview of this Guide

This guide describes how to set up Opsware SAS features after an Opsware core has been installed in a facility. Specifically, this guide describes how to set up users, user groups, administrators, customers, and facilities. It discusses how to set up Opsware SAS features, such as OS Provisioning, Software Management, and Code Deployment and Rollback, and how to manage packages in the Software Repository.

Contents of this Guide

This guide contains the following chapters and appendices:

Chapter 1: Server Management Configuration: provides information about how you can control server management in your operational environment by creating customers to associate servers with, creating and modifying server attributes to categorize the servers running in your operational environment, and by creating IP address groups to control which customers your servers are associated with when you install the Opsware Agent on them.

Chapter 2: Software Management Policies: Provides information about creating and managing software policies, managing folders, and managing packages.

Chapter 3: OS Provisioning Setup: Provides a description of all tasks necessary to prepare for operating system provisioning including media management, operating system specific tasks, OS installation profiles, build customization scripts, OS build process default definitions, operating system definitions in templates, and details of hardware support.

Chapter 4: Code Deployment Setup: Provides information about setting up services, synchronizations, and sequences in the Code Deployment and Rollback (CDR) feature to deploy code and content to managed servers.

Chapter 5: Visual Packager: Provides information about creating software packages from managed server information, and includes such topics as how to configure the SAS Client to access a packaging server, how to create a package, and explains different methods of software packaging.

Appendix A: CML Tutorial & Reference: Explains and illustrates the basics of Opware's Configuration Markup Language (CML) through a tutorial, and provides basic reference information.





Conventions in this Guide

This guide uses the following typographical and formatting conventions.

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

Icons in this Guide

This guide uses the following icons.

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

Guides in the Documentation Set and Associated Users

- The *Opware[®] SAS User's Guide: Server Automation* is intended for system administrators responsible for all aspects of managing servers in an operational environment. It describes how to use Opware SAS, introducing the system and the user interface. It provides information about managing servers, remediating servers, script execution, configuration tracking, deploying and rolling back code, and agent deployment. It also explains how to use the Opware Global Shell and open a Remote Terminal on managed servers.
- *Opware[®] SAS User's Guide: Application Automation* is intended for system administrators responsible for performing the day-to-day functions of managing servers. It reviews auditing and compliance, software packaging, visual application

management, application configuration, and software and operating system installation on managed servers.

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

Opsware, Inc. Contact Information

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

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

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

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

For troubleshooting information, see the Opsware Knowledge Base:

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

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

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

Chapter 1: Server Management Configuration

IN THIS CHAPTER

This section discusses the following topics:

- Ways to Configure Server Management
- Customer Accounts in Opware SAS
- Customer Account Administration
- Server Attributes
- IP Range Groups and IP Ranges

Ways to Configure Server Management

Opware SAS includes several ways to control the ways that servers are managed in your operational environment:

- Creating customers in Opware SAS and associating servers in the operational environment with those customers.
- Setting up IP range groups and IP ranges so that servers are automatically associated with customers when Opware users install the Opware Agent on servers running in the operational environment
- Creating and modifying the values for Server Attributes so that Opware users can identify broad categories of servers and what they are used for, as well as to describe their various stages of life cycle deployment.

Supported Operating Systems for Managed Servers

This section lists the supported operating systems for Opware Agents and the SAS Client.

The following table lists the supported operating systems for Opware Agents, which run on the servers managed by Opware SAS.

Table 1-1: Opware Agent Supported Operating Systems

SUPPORTED OPERATING SYSTEMS FOR OPSWARE AGENT	VERSIONS	ARCHITECTURE
AIX	AIX 4.3 AIX 5.1 AIX 5.2 AIX 5.3	POWER POWER POWER POWER
HP-UX	HP-UX 10.20 HP-UX 11.00 HP-UX 11.11 HP-UX 11.23 (11i v2)	PA-RISC PA-RISC PA-RISC PA-RISC and Itanium
Sun Solaris	Solaris 6 Solaris 7 Solaris 8 Solaris 9 Solaris 10 (Update 1, Update 2, Update 3)	Sun SPARC Sun SPARC Sun SPARC Sun SPARC Sun SPARC, 64 bit x86, 32 bit x86 and Niagara
Fujitsu Solaris	Solaris 8 Solaris 9 Solaris 10	Fujitsu SPARC Fujitsu SPARC Fujitsu SPARC
Windows	Windows NT 4.0 Windows 2000 Server Family Windows Server 2003 Windows XP Professional	32 bit x86 32 bit x86 32 bit x86 and 64 bit x86 32 bit x86

Table 1-1: Opware Agent Supported Operating Systems (continued)

SUPPORTED OPERATING SYSTEMS FOR OPSWARE AGENT	VERSIONS	ARCHITECTURE
Red Hat Linux	Red Hat Linux 7.3	32 bit x86
	Red Hat Linux 8.0	32 bit x86
	Red Hat Enterprise Linux 2.1 AS	32 bit x86
	Red Hat Enterprise Linux 2.1 ES	32 bit x86
	Red Hat Enterprise Linux 2.1 WS	32 bit x86
	Red Hat Enterprise Linux 3 AS	32 bit x86 and 64 bit x86 and Itanium
	Red Hat Enterprise Linux 3 ES	32 bit x86 and 64 bit x86 and Itanium
	Red Hat Enterprise Linux 3 WS	32 bit x86 and 64 bit x86 and Itanium
	Red Hat Enterprise Linux 4 AS	32 bit x86 and 64 bit x86
	Red Hat Enterprise Linux 4 ES	32 bit x86 and 64 bit x86
	Red Hat Enterprise Linux 4WS	32 bit x86 and 64 bit x86
	Red Hat Enterprise Linux Server 5	32 bit x86 and 64 bit x86
Red Hat Enterprise Linux Desktop 5	32 bit x86 and 64 bit x86	
SUSE Linux	SUSE Linux Enterprise Server 8	32 bit x86
	SUSE Linux Standard Server 8	32 bit x86
	SUSE Linux Enterprise Server 9	32 bit x86 and 64 bit x86
	SUSE Linux Enterprise Server 10	32 bit x86 and 64 bit x86
VMware	ESX Server 3	32 bit x86 and 64 bit x86



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

The following table lists the operating systems supported for the SAS Client.

Table 1-2: SAS Client Supported Operating Systems

SUPPORTED OPERATING SYSTEMS FOR SAS CLIENT	VERSIONS	ARCHITECTURE
Windows	Windows Vista	32 bit x86 and 64 bit
	Windows XP	x86
	Windows 2003	32 bit x86
	Windows 2000	32 bit x86
		32 bit x86

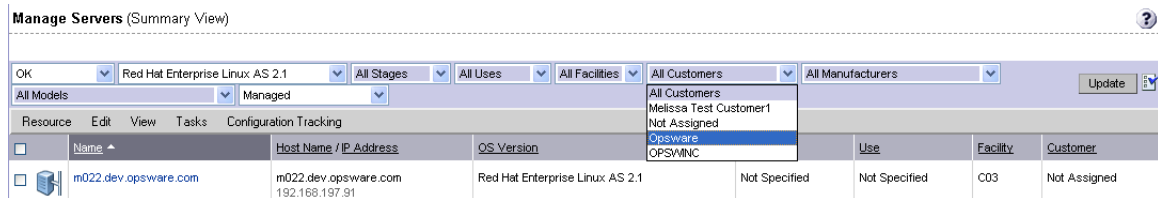
Customer Accounts in Opware SAS

Many enterprise customers have consolidated disparate IT operations into a single operation, yet they still need separate reporting, billing, and management for different business units or groups (for example, West Coast Office, East Coast Office, and London Office).

Opware SAS accommodates these requirements. Within the SAS Web Client, Opware users perform server provisioning and management by using customer accounts.

When an Opware administrator creates a customer in Opware SAS, a value for that customer is automatically added to the customer filter in the Managed Servers list, as Figure 1-1 shows.

Figure 1-1: Customer Filter in the Managed Servers List



By using customer accounts in the SAS Web Client, you can segregate servers that belong to different business units. By segregating servers, you can have separate accounting for each customer or different levels of security for different customers. You might want to segregate the servers based on the department or business unit to which they belong for many reasons.

By default, Opware SAS is shipped with the following two customers:



- **Customer Independent:** A global customer in Opware SAS. Resources (applications, patches, and templates) that are associated with “Customer Independent” can be installed on any managed server, no matter what customer it is associated with.
- **Not assigned:** The servers are not associated with a customer. You can install applications, patches, or templates that are Customer Independent on Not Assigned servers. However, you cannot install or use any resources associated with a customer on a server that is not assigned to a customer.

When you assimilate a server into Opware SAS, the server is associated with the Not Assigned customer if IP ranges were not created to automatically associate assimilated servers with customers. See Figure 1-2.



Opware Inc. recommends that you associate servers with customers, if necessary, by using the Server Properties pages. See the *Opware® SAS User's Guide: Server Automation* for more information on editing the properties of a server.

Figure 1-2: Customers List Under Environment in the SAS Web Client

Customers		Customers	
	Name		Name
	12204		Corp Test
	Big Corp		Big Corp2
	Test Cust		Customer Independent
	E-Commerce		Not Assigned

Associated Servers with Customers

An Opsware user or an Opsware administrator can set up an IP range group so that servers are automatically associated with customers when users perform the following server management tasks:

- Manage servers running in the operational environment by installing an Opsware Agent on the servers

See the *Opsware® SAS User's Guide: Server Automation* for more information on how to install an Opsware Agent.

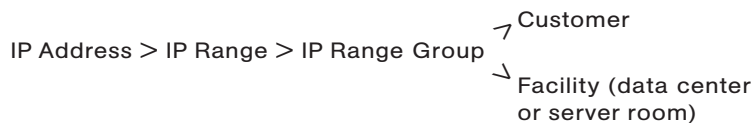
- Use the OS Provisioning feature to install operating systems on bare-metal servers

See the *Opsware® SAS User's Guide: Application Automation* for more information on OS provisioning.

To set up this automatic customer association, you must create IP range groups for customers and specify the ranges of IP addresses that the groups contain.

In the SAS Web Client, an IP range group is both a physical and logical list – an accounting way to group ranges of IP address and assign them to a particular customer. An IP range identifies a range of IP addresses within an IP range group.

When you set this up, IP addresses get their customer association through the IP range, which, in turn, gets its customer association from the IP range group.



See “IP Range Groups and IP Ranges” on page 41 in this chapter for more information.

The loose relationship between server and IP address means that you can associate a server with a different customer from its IP address.

Even when IP range groups are set up for a customer, a server's IP address does not necessarily determine the customer to which the server is associated because a user can change the customer association in the Server Properties page.

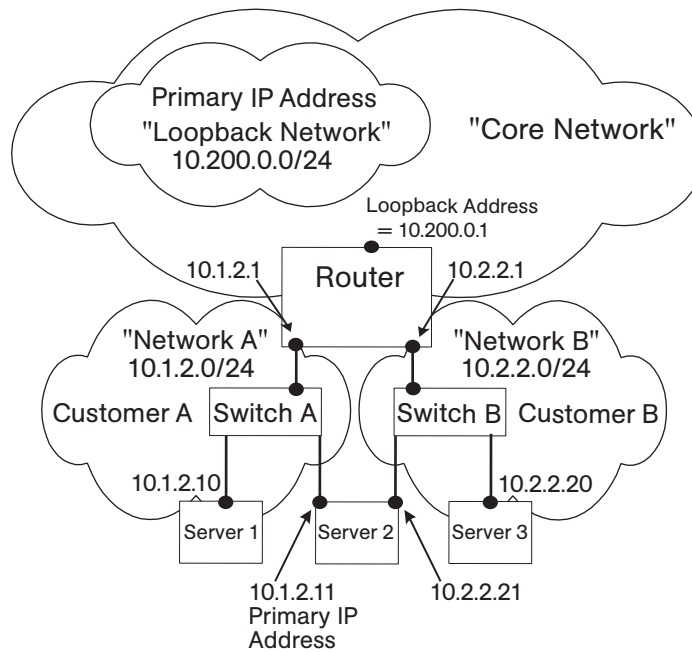
See *Opsware® SAS User's Guide: Server Automation* for information about how to change the customer association for a server.

The customer association for a server is based on the management IP address of the server and not the primary IP address.

See the *Opsware[®] SAS User's Guide: Server Automation* for more information about how Opsware SAS uses management IP addresses for servers.

However, a server always belongs to the same facility (data center or server room) as its primary IP address. Opsware SAS enforces the relationship between server and facility at hardware registration. See Figure 1-3.

Figure 1-3: Primary IP Addresses in Opsware SAS



In this illustration, the following conditions apply:

- Server 1 belongs to Customer A.
- Server 2 belongs to Customer A but has IP addresses in Network A and Network B.
- Server 3 belongs to Customer B.
- The Router belongs to the Core Network but has IP addresses in Network A and Network B.

Customer Account Administration

This section provides information about customer account administration within Opware SAS and contains the following topics:

- As an Opware administrator, you can add customers or update information and configuration settings for an existing customer.
- Creating a Customer
- Updating Customer Information and Settings
- Setting Custom Attributes for Customers
- Restrictions for Deleting Customers
- Deleting a Customer

As an Opware administrator, you can add customers or update information and configuration settings for an existing customer.

Creating a Customer





As an Opware administrator, you can add customers to your Opware SAS installation to create designations by business unit to provide management of Opware SAS operations and configuration.

Perform the following steps to create a customer:

- 1** From the navigation panel, click Environment ► Customers.

The Customers page displays a list of all current customers, as Figure 1-4 shows.

Figure 1-4: List of Current Customers on Customers Page

Customers			
Delete		New Customer	
	Name		Name
<input type="checkbox"/>	 12204		Customer Independent
<input type="checkbox"/>	 E-Commerce		Not Assigned

- 2** Click **New Customer**.

The Customer: New Customer page appears where you can define the settings for a customer account, as Figure 1-5 shows.

Figure 1-5: Information Section of New Customer Page

Customers: New Customer	
Return to Customers	
New Customer	
Information	
Name:	<input type="text"/>
Short Name:	<input type="text"/>
	Must consist of uppercase letters, numbers, '-', and '_'.
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

- 3 In the Information section, enter a Name for the customer and a short name, or a nickname by which the customer name will display.

The short name is limited to 25 characters and must consist of uppercase letters, numbers, hyphens (-), and underscores (_).

- 4 If you have more than one facility, specify the associated facilities for the customer. To add a facility to the customer, select a facility from those displayed in the Available Facility list and click the left arrow.



You only see the fields to add a facility for a customer when Opware SAS is running in multiple facilities.

- 5 When you finish defining the customer, click **Save**. A confirmation message appears that says that the customer was successfully created.



By default, no access permission of this customer is granted to any user groups. In order to permit users to access this customer, you must grant access permissions to the appropriate user groups.

- 6 Click **Continue**. The Manage Customer page appears.



You must log out and log in again to see the updated Customer list.

After you create a customer, you can define custom attributes for the customer. These custom attributes include setting miscellaneous parameters and named data values. Users can write scripts that use these parameters and data values when they perform a variety of functions, including network and server configuration, notifications, and CRON script configuration.

See “Setting Custom Attributes for Customers” on page 32 in this chapter for more information

Updating Customer Information and Settings

As an Opware administrator, you can update information or change configuration settings for existing customer accounts defined for your Opware SAS.

Perform the following steps to update an existing customer:

- 1 From the navigation panel, click Environment ► Customers. The Customers page appears, which shows a list of your existing customer accounts.
- 2 Click the hyperlinked name of the customer whom you want to update.

The Customers: Edit Properties page appears, as Figure 1-6 shows. The list box on the left side includes facilities assigned to the customer. The list box on the right side includes all other Opware SAS-managed facilities that are available to be added for a new or existing customer.

To change the name for the customer who appears in the SAS Web Client, edit the name that appears in the Name field.

Figure 1-6: Assign Facility Section of the Customers: Edit Properties Page

The screenshot displays the 'Customers: Edit Properties' interface. At the top, there is a 'Return to Customers' link. Below it are two tabs: 'Properties' (selected) and 'Custom Attributes'. The main section is titled 'Edit Customer' and contains an 'Information' table with the following details:

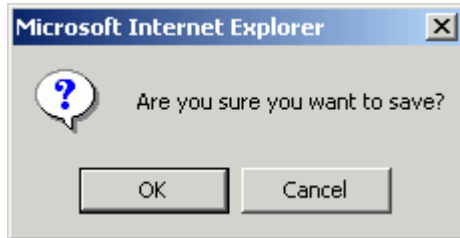
ID:	1220007
Name:	<input type="text" value="Important Customer"/>
Short Name:	VIP
Status:	ACTIVE

Below the table is the 'Assign Facility' section, which consists of two lists: 'Assigned Facility' (currently empty) and 'Available Facility' (containing 'C07'). Between these lists are two arrow buttons: a right-pointing arrow (→) and a left-pointing arrow (←). At the bottom of the form are 'Save' and 'Cancel' buttons.

- 3 To add a facility to the customer, select a facility from those displayed in the Available Facility list and click the left arrow.
- 4 Click **Save**.

A message appears that confirms that you want to save the changes you just made, as Figure 1-7 shows.

Figure 1-7: Confirm Save Message



- 5 Click **OK** to save the changes.

The Customers page appears, which allows you to continue making changes to customer properties.

Setting Custom Attributes for Customers

You can use the Custom Attribute function to apply special properties to customers.

Perform the following steps to apply special properties to customers:

- 1 From the navigation panel, click Environment ► Customers. The Customers page appears, which shows a list of your existing customer accounts.
- 2 Click the hyperlinked name of the customer who you want to update.
The Customers: Edit Properties page appears.
- 3 Select the Custom Attributes tab.
The Customers: Edit Custom Attributes page appears. If custom attributes have previously been applied to this customer, they appear on this page.
- 4 Click **New**.

The SAS Web Client displays the Customers: New Custom Attributes page, as Figure 1-8 shows.

Figure 1-8: New Custom Attributes Page

Customers: New Custom Attribute	
Return to Edit Custom Attributes	
Edit a Name and Value suitable for the new custom attribute.	
Name:	<input type="text"/>
Value:	<input type="text"/>

These named values are used to provide parameters to Opware SAS, for example, to customize displays or provide settings to use during installation or configuration of packaged software in the operational environment.



Do not use either an asterisk (*) or a question mark (?) in the name field.



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

- When you finish entering names and values, click **Save**, or exit without entering any values by clicking the Return to Customers link.

Restrictions for Deleting Customers

You can only delete a customer account from the SAS Web Client when the following conditions are true for the customer:

- No Nodes are attached to the customer account.
- The customer account does not own any software packages; the packages uploaded for the customer must be deleted or deprecated.

- All servers assigned to the customer are deactivated.
- No IP Range Groups are created for the customer.
- No IP Ranges are created for the customer.
- No server groups are created for the customer.

If any of these restrictions apply to this customer, a message appears and you cannot delete the customer.

If the restrictions do not apply, the user is prompted to move deactivated servers to the Not Assigned customer account.

Deleting a Customer

Deleting a customer removes the customer information from Opware SAS and moves deactivated servers assigned to the customer to the Not Assigned customer account or deletes the data about the servers from the Model Repository database.

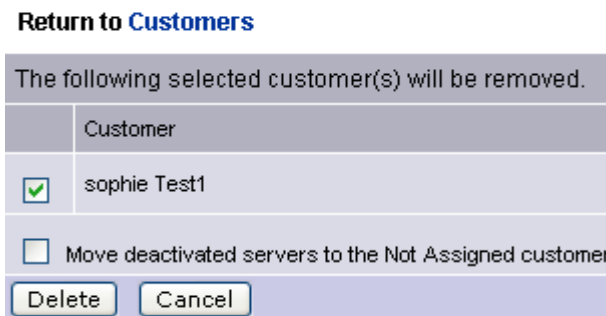
See “Restrictions for Deleting Customers” on page 33 in this chapter for more information

Perform the following steps to delete a customer:

- 1** From the navigation panel, click Environment ► Customers. The Customers page appears, which shows a list of your existing customer accounts.
- 2** Select the check box next to the customer whom you want to delete.
- 3** Click **Delete**.

A confirmation page appears, which verifies that the selected customer will be deleted, as Figure 1-9 shows.

Figure 1-9: Customers: Delete Confirmation Page



- 4** Click **Delete**.

A confirmation page appears, as Figure 1-10 shows.

Figure 1-10: Delete Customer: Confirmation Page



- 5 Click **Continue**.

The SAS Web Client displays the updated Customers page.

Server Attributes

This section provides information about server attributes within Opsware SAS and contains the following topics:

- Associated Servers with Customers
- Creating Server Use Values
- Editing Server Use Values
- Deleting Server Use Categories
- Creating Deployment Stage Values
- Editing Deployment Stage Values
- Deleting Deployment Stage Values

The Server Attribute function is used to identify broad categories of servers and what they are used for, as well as to describe their various stages of life cycle deployment.

Opsware SAS comes with three Server Use categories already defined (and which cannot be changed or deleted): Not Specified, Production, and Staging. It also has a Deployment Stage category pre-defined - Not Specified - (which also cannot be changed or deleted).

The attributes defined here, along with the default attributes, populate two lists in the Server Management function: Server Use and Deployment Stage. See the *Opsware[®] SAS User's Guide: Server Automation* for more information.

Creating Server Use Values

Perform the following steps to create server use values:

- 1 From the navigation panel, select Administration ► Server Attributes. The Server Attributes page appears with the Server Use tab displayed, which shows all previously defined server use categories, as Figure 1-11 shows.

Figure 1-11: Server Use Tab of the Server Attributes Function

Server Attributes			
Server Use		Deployment Stage	
Delete		New Value	
<input type="checkbox"/> Name	Code Deployment	Description	
<input type="checkbox"/> DevLab	Enabled	Development Lab Servers	
<input type="checkbox"/> DevLab2	Enabled	Development Lab, South Building	
<input type="checkbox"/> Development	Disabled	Site development	

- 2 Click **New Value**. The Create Server Use Value page appears, as Figure 1-12 shows.

Figure 1-12: Create Server Use Value Page

[Return to Server Use](#)

Name:

Description:

Code Deployment:

- 3 Enter the name of the Server Use category that you want to create. This name appears in the Server Use list in the Managed Servers area of the system. This field is required.
- 4 Enter a description of the server use value that you are defining.
- 5 Select the Code Deployment check box if you want this server use category to also appear in the Code Deployment list.
- 6 Click **Save**.

Editing Server Use Values

Perform the following steps to edit server use values:

- 1 From the navigation panel, select Administration ► Server Attributes. The Server Attributes page appears with the Server Use tab displayed, which shows all previously-defined server use categories.
- 2 The names of the server use categories that you already defined, as well as the default categories names, are hyperlinks. Click the link to edit the values of the categories. The Edit Server Use Value page appears, as Figure 1-13 shows.

Figure 1-13: Edit Server Use Value Page

Edit Server Use Value	
Return to Server Use	
Name:	<input type="text" value="DevLab"/>
Description:	<input type="text" value="Development Lab Servers"/>
Code Deployment:	<input checked="" type="checkbox"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

For the default categories of Not Specified, Production, and Staging, only the description can be modified.

For the server use categories that you have defined, all values can be modified.

- 3** Make any necessary changes to the Name, Description, or Code Deployment fields.
- 4** Click **Save**.

Deleting Server Use Categories

Perform the following steps to delete the server use categories:

- 1** From the navigation panel, select Administration ► Server Attributes. The Server Attributes page appears with the Server Use tab displayed, which shows all previously-defined server use categories.
- 2** Select the check box next to each of the server use categories that you want to delete. You cannot delete the ones with no check boxes - Not Specified, Production, and Staging.
- 3** Click **Delete**. A confirmation window appears. You can view or hide the details of the server uses that you are about to delete.
- 4** Click **Delete**. The server use values are deleted, and the Server Attributes page refreshes, which shows the remaining server use values.

Creating Deployment Stage Values

Perform the following steps to create deployment stage values:

- 1** From the navigation panel, select Administration ► Server Attributes. The Server Attributes page appears with the Server Use tab displayed, which shows all previously-defined server use categories.

- 2 Select the Deployment Stage tab. The list of previously defined deployment stages appears, as Figure 1-14 shows.

Figure 1-14: Deployment Stage Tab of the Server Attributes Function

Server Use		Deployment Stage
<input type="checkbox"/> Delete		<input type="button" value="New Value"/>
<input type="checkbox"/> Name	Description	
<input type="checkbox"/> Decommissioned	Site has been completely shut down and is no longer in use.	
<input type="checkbox"/> In Deployment	Server is actively being built.	
<input type="checkbox"/> Live	Code is successfully deployed and tested, site is ready for live traffic.	
<input type="checkbox"/> Not Specified	The stage of the server is not yet known.	
<input type="checkbox"/> Offline	Server may still be operating, but has been removed from active management.	
<input type="checkbox"/> Ops Ready	Build tasks have been completed and server is ready for Ops Ready testing.	

- 3 Click **New Value**. The Create Deployment Stage Value page appears, as Figure 1-15 shows.

Figure 1-15: Create Deployment Stage Value Page

Create Deployment Stage Value

[Return to Deployment Stage](#)

Name:	<input type="text"/>
Description:	<input style="height: 60px;" type="text"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

- 4 Enter the name of the deployment stage. This field is required.

- 5 Enter a description of the deployment stage.
- 6 Click **Save**.

Editing Deployment Stage Values

Perform the following steps to edit deployment stage values:

- 1 From the navigation panel, click Administration ► Server Attributes. The Server Attributes page appears with the Server Use tab displayed, which shows all previously defined server use categories.
- 2 Select the Deployment Stage tab. The list of previously defined deployment stages appears.
- 3 Each of the deployment stages on the list is a hyperlink. Click the name of the deployment stage whose values you want to edit. The Edit Deployment Stage Value page appears, as Figure 1-16 shows.

Figure 1-16: Edit Deployment Stage Value Page

Edit Deployment Stage Value	
Return to Deployment Stage	
Name:	<input type="text" value="Decommissioned"/>
Description:	<input type="text" value="Site has been completely shut down and is no longer in use."/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

- 4 Change the Name or Description as necessary.
- 5 Click **Save**.

Deleting Deployment Stage Values

Perform the following steps to delete deployment stage values:

- 1 From the navigation panel, select Administration ► Server Attributes. The Server Attributes page appears with the Server Use tab displayed.

- 2** Select the Deployment Stage tab. The list of previously defined deployment stages appears.
- 3** Select the check box next to the deployment stage that you want to delete. The one with no check box, Not Specified, cannot be deleted.
- 4** Click **Delete**. A confirmation page appears. You can view or hide the details of the deployment stages that you are about to delete.
- 5** Click **Delete**. The Server Attributes page with the Deployment Stage tab appears, which shows the remaining deployment stages.

IP Range Groups and IP Ranges

This section provides information on IP range groups and IP ranges within Opware SAS and contains the following topics:

- Overview of IP Range Groups and IP Ranges
- Creating an IP Range Group
- Creating an IP Range
- Changing Address Ranges on IP Ranges
- Increasing and Decreasing the Prefix Length
- Changing the Status of an IP Address in an IP Range

Overview of IP Range Groups and IP Ranges

An Opware user or an Opware administrator can set up IP range groups and IP ranges so that servers are automatically associated with customers when users perform the following server management tasks:

- Manage servers running the operational environment by installing an Opware Agent on the servers

See the *Opware[®] SAS User's Guide: Server Automation* for more information on installing an Opware Agent on a server.

- Use the OS Provisioning feature to install operating systems on bare-metal servers

If you do not assign an IP range group to a customer, by default, a server is not assigned to a customer (Not Assigned appears in the Customer column of the server list) when you install an Opware Agent on the server.

An IP Range Group is a group of IP ranges that belong to a customer. It is both a physical and logical list – an accounting way to group IP ranges and assign them to a specific customer.

In the SAS Web Client, an IP range identifies a range of IP addresses (in the OSI model – layer 3 IP address ranges). Each IP range can contain many IP addresses. The range of IP addresses is dependent on the subnet specified.

There is no direct association of an IP range with a specific customer; an IP range inherits its association to a customer from the IP Range Group it is created in.

See “Customer Account Administration” on page 28 in Chapter 1 for more information for more information about Associated Servers with Customers.

Several types of IP Ranges are available in the SAS Web Client, as Figure 1-17 shows.

Figure 1-17: Types of IP Ranges in the SAS Web Client

IP Ranges: Create IP Range Type Customer UNKNOWN Facility"Folsom Data Center (core0)"			
Return to IP Ranges			
IP Range 1			
IP Range Name:	<input type="text"/>	IP Range type:	Please Select IP Range Type ▾
IP Range Group:	Please Select IP Range Group ▾	Pool Description:	Please Select IP Range Type
Sub-Type:	Please Select Sub Type ▾	Subnet/CIDR:	CONSOLE CORE DMZ PUBLIC VPN WAN
Pool Name:	<input type="text"/>		

Creating an IP Range Group

You perform this task to create a group of IP ranges for a specific customer. After you create the group, you can designate the IP ranges that you want in that group.

Perform the following steps to create an IP Range Group:

- 1 From the navigation panel, click Environment ► IP Range Groups. The IP Range Groups page appears, as Figure 1-18 shows.

Figure 1-18: IP Range Groups Page in the SAS Web Client

<input type="checkbox"/>	Name	Customer
<input type="checkbox"/>	CORP	Opware
<input type="checkbox"/>	Default	Not Assigned

- 2 From the list, select the facility in which you want to create the IP range group and click **Update**. The list of IP range groups for that facility appears.
- 3 Click **New** at the top of the page. The IP Range Groups: Create IP Range Group page appears.
- 4 Enter a name for the new IP range group.
- 5 Select the customer from the drop-down list.
- 6 Click **Save**.

Creating an IP Range

Perform the following steps to create an IP Range:

- 1 From the navigation panel, click Environment ► IP Ranges. The IP Ranges: View IP Ranges page appears, as Figure 1-19 shows.

Figure 1-19: IP Ranges for the Default Customer ("Not Assigned")

IP Ranges: View IP Ranges						
IP Ranges		IP Range Types				
Not Assigned		C07		Update		
New						
Click on IP Range name to view and edit details. Click on Subnet/CIDR to change the CIDR value.						
Default (Not Assigned)						
<input type="checkbox"/>	IP Range Name	Pool Name	Description	IP Range Type	Sub-Type	Subnet / CIDR
	Default	Default	Holding pool for IPs used by Devices but not managed as part of other VLANs	PUBLIC	PRODUCTION	n/a/-1
Delete		selected IP Ranges				

- 2 From the list, select the customer and facility in which you want to create the IP range and click **Update**. The list of IP ranges for that customer and facility appears.
- 3 Click **New** at the top of the page. The IP Ranges: Create IP Range Type page appears, as Figure 1-20 shows. You can add up to five new IP ranges at a time.

Figure 1-20: Creating an IP Range

IP Ranges: Create IP Range Type Customer UNKNOWN Facility "C07"	
Return to IP Ranges	
IP Range 1	
IP Range Name:	<input type="text"/>
IP Range type:	Please Select IP Range Type
IP Range Group:	Please Select IP Range Group
Pool Description:	<input type="text"/>
Sub-Type:	Please Select Sub Type
Subnet/CIDR:	<input type="text"/> <input type="text"/>
Pool Name:	<input type="text"/>

- 4** Define the following properties for each IP range:
 - **IP Range Name:** For example, VLAN999 or SERVER100.
 - **IP Range Group:** A customer might have several IP range groups and you must select one for the IP range that you are creating.
 - **Sub-Type:** For example, Development, Production, Staging, and so forth.
 - **Pool Name:** For example, SAMPLE CUSTOMER SERVER pool.
 - **IP Range Type:** For example, SERVER, PUBLIC, CONSOLE, TRANSIT, CORE, and so forth.
 - **Pool Description:** Provides detailed information about the IP range.
 - **Subnet:** For example, 10.2.0.0.
 - **Mask or Prefix Length:** Enter the prefix length or netmask (for example, 24, for / 24, a netmask 255.255.255.0) in the CIDR field.

You must complete all fields for each new IP range, which assumes that you have specific knowledge of your network's configuration and know the correct entries to include.

- 5** After you complete all entries, click **Save** at the bottom of the page.

Changing Address Ranges on IP Ranges

Classless Inter-Domain Routing (CIDR) in Opware SAS provides a way of specifying a range of IP addresses to include in an IP range.

Opware SAS might take several minutes to display an IP range with many IP addresses. For example, an IP Range with CIDR 19 (which has 8,192 IP addresses) might take 5 minutes to display in the SAS Web Client.

Perform the following steps to change address ranges on IP ranges:

- 1** From the navigation panel, click Environment ► IP Ranges. The IP Ranges: View IP Ranges page appears.
- 2** From the list, select the customer and facility whose IP range you want to update and click **Update**. The list of IP ranges for that customer and facility appears.

- Click the SUBNET/CIDR link at the end of the row for the IP range that you want to change, as Figure 1-21 shows.

Figure 1-21: IP Ranges in the SAS Web Client

IP Ranges: View IP Ranges						
IP Ranges		IP Range Types				
All Customers	C07	Update				
New						
Click on IP Range name to view and edit details. Click on Subnet/CIDR to change the CIDR value.						
CORP (Opsware)						
<input type="checkbox"/>	IP Range Name	Pool Name	Description	IP Range Type	Sub-Type	Subnet / CIDR
<input type="checkbox"/>	Some Range To Be Deleted	SOME POOL	This is <~!@#\$\$%^&*()_> some description	CONSOLE	DEVELOPMENT	10.0.9.0/24

The last two digits in the Subnet/CIDR column make up the current prefix length for a particular IP range. The IP Range: Change CIDR page appears.

- To change the current CIDR setting, select a new value from the list in the New CIDR column.
- Click **Change**.

Changing the prefix length in the SAS Web Client does not automatically change the net masks of servers for the servers themselves.

Increasing and Decreasing the Prefix Length

You can use the SAS Web Client to increase or decrease the length of an IP range.

Increasing the Prefix Length

Increasing the prefix length reduces the IP range size. For example, if you have an IP range with prefix length 24 and it has 256 IP addresses in it, changing the prefix length to 25 results in the creation of two IP ranges with prefix length 25, each containing 128 IP addresses.

Example:

Network A - 10.1.0.0/24

Becomes:

Network A - 10.1.0.0/25

Network B: 10.1.0.128/25 (new network)

Decreasing the Prefix Length

Decreasing the prefix length expands the IP range size. Take the two CIDR 25 IP ranges from above. On the first IP range, changing the prefix length to 24 results in one IP range that contains twice as many IP addresses as before. The two original CIDR 25 IP ranges are combined to make one larger CIDR 24 IP range.

Example:

Network A - 10.1.0.0/24

Network B - 10.1.1.0/24

Becomes:

Network AB: 10.1.0.0/23 (one network)



This change only works if the two CIDR 25 IP ranges occupy contiguous blocks in the same IP range group. If they do not occupy contiguous blocks, you get an error message.

Changing the Status of an IP Address in an IP Range

You can use the IP Range feature to change the status of that IP address in the SAS Web Client. For example, you might want to reserve an available IP address because you will assign it to a specific server in the next few days.

The status of an IP address automatically changes from available to assigned when a server with that IP address registers its hardware with Opware SAS.

Perform the following steps to change the status of an IP address in an IP range:

- 1** From the navigation panel, click Environment ► IP Ranges. The IP Ranges: View IP Ranges page appears.
- 2** From the list, select the customer and facility for which you want to assign IP addresses and click **Update**. The list of IP ranges for that customer and facility appears.
- 3** Click the name for the IP range in which you want to assign IP addresses. The IP Range: View IP Range page appears. By default, the View tab displays.

The bottom of the page contains the IP addresses within that range. For each assigned or reserved IP address in the range, you can see its status.

- 4** Click an individual IP address link.

From the page that appears, you can change the status of the IP address (to ASSIGNED, AVAILABLE, RESERVED, and so forth). See Figure 1-22.

Figure 1-22: Editing the Properties of an IP Address in an IP Range

IP Range Groups: Edit IP	
Return to View IP Range	
Edit IP 192.168.8.141	
IP Address:	192.168.8.141
Status:	<input type="text" value="NETWORK"/> <ul style="list-style-type: none"> ASSIGNED AVAILABLE NOT-AVAILABLE RESERVED NETWORK DHCP BROADCAST GATEWAY VIRTUAL <li style="background-color: #800080; color: white;">NETWORK

- 5** Select the status from the list. IP addresses can have one of the following statuses:
- **ASSIGNED:** A server is registered with this IP address.
 - **AVAILABLE:** Available IP address.
 - **NOT AVAILABLE:** Used to reserve an IP address for future use. For example, you might want to build a new server but need an IP address for the server prior to it being plugged into the network. Setting the status of an IP address to NOT AVAILABLE reserves it, so that another user does not take that IP address before the server is racked, stacked, and plugged into the network.
 - **RESERVED:** The first couple of IP addresses after the first IP address is reserved.
 - **NETWORK:** Always assigned to the first IP address in a subnet.
 - **DHCP:** IP addresses reserved for use by a DHCP server.
 - **BROADCAST:** A special IP address reserved for sending a message to all stations.
 - **GATEWAY:** An IP address that acts as an entrance to another network.
 - **VIRTUAL:** Indicates a virtual IP address, such as `www.samplecustomer.com`, which is an IP address associated with a load balancer. The IP address does not correspond to any server, but yet the active load balancer responds to this request and forwards it to the appropriate Web server.

6 Click **Save**.

Chapter 2: Software Management Setup

IN THIS CHAPTER

This section contains the following topics:

- Overview of Software Management
- Software Management Process
- Software Management Setup Tasks
- Library
- Overview of Software Policies
- Folders
- Overview of Package Management
- RPM Deployment

Overview of Software Management

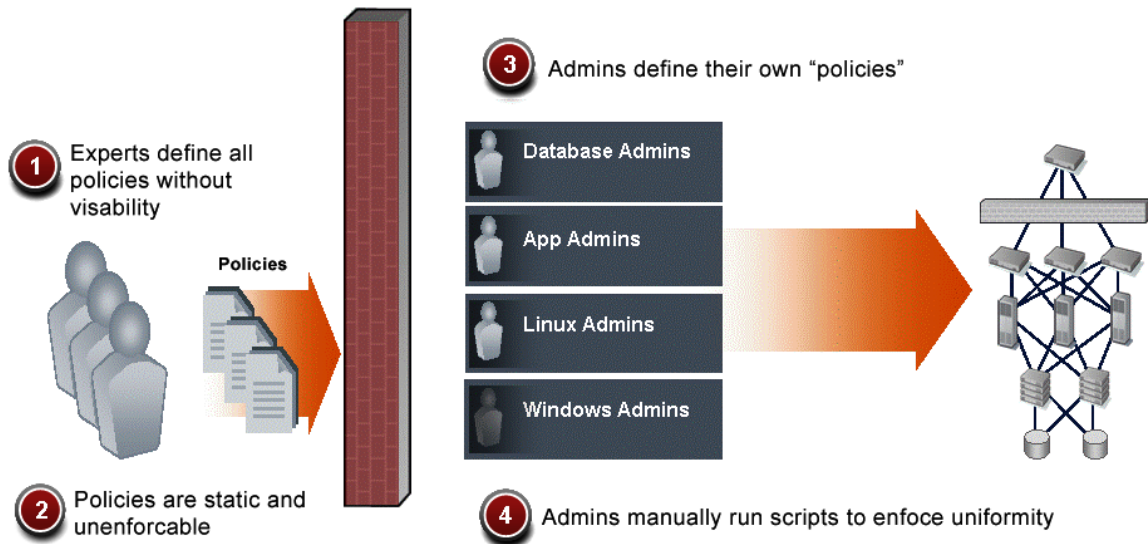
The Software Management feature in Opsware SAS provides a powerful mechanism to model software by using software policies to automate the process of installing software and configuring applications on a server in a single step. In addition, the Software Management feature provides a structure to organize your software resources in folders and define security permissions around them. This feature allows you to verify the compliance status of a server and remediate non-compliant servers.

Policy-Based Software Management

One of the most difficult problems in the IT organizations today is the deployment of software to servers running in the operational environment. Many IT organizations define policies for creating systems in their operational environment and set standards for server

configuration. Often, policies are written by subject-matter experts, like the application developer or security administrator in an IT organization, or policies are leveraged from industry standards organizations as shown in Figure 2-1.

Figure 2-1: Defining Policies for Software Deployment.



Clearly defining policies is the correct approach to managing software deployment. However, these policies are typically static and unenforceable because the policy creators have no means to enforce their policies. Opware SAS solves the software deployment problem by introducing the Software Management feature in the SAS Client.

Using the Software Management feature allows IT managers to enforce policies, standardize operations, and ensure compliance against defined software policies. In the SAS Client, policy setters create a model of their IT environment by using a software policy. Defining a software policy is like defining a server baseline to ensure that all servers are provisioned on a standardized basis with the right content.

In Opware SAS, policy setters create a software policy that specifies the packages and patches to be installed, and the configurations to be applied to the managed servers in their IT environment. System administrators can then manage the servers in their environment by applying the software policy to the servers. Opware SAS applies the changes to the managed servers when you remediate the managed servers with the software policies.

When a change needs to be made to a software policy, a policy setter simply changes the baseline defined in the software policy and the incremental differences are applied across the target servers. This automated and systematic method for keeping large numbers of servers in compliance eliminates the need to store and manage hundreds of rigid images and provides the means to easily restore or rollback servers to a previous working state.

Software Management Features

The Software Management feature in the SAS Client enables you to perform the following functions:

- **Create an organizational structure for software**

The folder hierarchy provides a way to organize your software resources. Folders act as containers for packages, software policies, and OS sequences.

- **Define security boundaries for folders**

Folders allow you to define security permissions to control access to their contents across user groups. You can set folder permissions to determine which user groups can view, use, and modify items within a folder.

- **Define a model-based approach to manage the IT environment in your organization**

Opware SAS enables a policy setter to create a model of their IT environment by using a software policy. In a software policy, the policy setter can specify the packages and patches to be installed, and the configurations to be applied to the managed servers in their IT environment. A system administrator can then manage the servers in their environment by applying the software policy to the servers. Opware SAS applies the changes to the managed servers when you remediate the managed servers with the model.

- **Enable sharing of software resources among user groups**

A software policy can contain various software resources managed by different user groups and located in different folders, thus allowing software resources to be shared across different groups.

- **Install and configure applications simultaneously**

Software policies can contain packages, patches, and application configurations which allow you to install the software and apply configurations to multiple servers in a single step.

- **Deploy multiple application instances on one server**

The Software Management feature allows you to install multiple instances of an application on a server by using relocatable ZIP packages.

- **Establish a software installation process**

The Software Management feature allows you to separate the different stages (download, installation, and reboot) of the software installation process. You can only independently schedule the various stages of the software deployment process. You can choose to get notified of job status via email upon successful completion of a stage and associate a Ticket ID with each job.

- **Verify compliance status of servers to software policies**

The Compliance Dashboard allows you to view the compliance state of a software policy to determine if a server is configured correctly and to remediate non-compliant servers.

- **Generate reports**

The Reporting feature allows you to generate reports that provide a summary of the software policy compliance across servers. You can generate reports that provide information about software policies on a given server.

- **Comprehensively search for software resources and servers**

Using the search functionality in the SAS Client, you can search for servers, software policies, folders, application configurations, patches and software, and perform actions on the search results.

Software Management Process

The software management process consists of the following key phases:

- **Defining security permissions**

In this phase, an Opware administrator assigns Folder permissions, Client feature permissions, and Customer constraints to define the security boundaries across various

user groups. By assigning folder permissions, the Opsware administrator can specify which user groups can view, use, and modify the software resources in a folder. And by assigning SAS Client feature permissions, the Opsware administrator can specify the actions the users in a user group can perform with the SAS Client.

See the *Opsware[®] SAS Administration Guide* for more information about defining security permissions.

- **Setting up folders and software policies**

In this phase, the policy setter, performs the set up tasks required for installing software. The set up tasks include uploading packages and patches to Opsware SAS, creating application configurations, setting up software policies with the software resources that are required to be installed, and managing dependencies between software resources across software policies. See “Software Management Setup Tasks” on page 55 in this chapter for more information.

- **Installing Software**

In this phase, a system administrator deploys the software to multiple servers by attaching software policies to managed servers and remediating the servers with the software policies. This phase includes tasks such as running software compliance scans to determine the compliance status of servers, remediating non-compliant servers, and generating software compliance reports across servers.

See *Opsware[®] SAS User's Guide: Application Automation* for more information about installing software.

Software Management Setup Tasks

The software management setup includes the following tasks:

- Package management tasks such as importing packages to Opsware SAS, and managing packages in Opsware SAS. See “Overview of Package Management” on page 93 in this chapter for more information.
- Patch management tasks such as importing patches to Opsware SAS, and managing patches and patch policies. See the *Opsware[®] SAS User's Guide: Application Automation* for more information about Patch Management.
- Folder management tasks such as creating folders, setting the folder hierarchy for your IT environment, and setting security boundaries using Folder permissions. See “Folders” on page 87 in this chapter for more information.

- Software policy management tasks such as adding software resources to a software policy, managing dependencies between software resources in a software policy, adding custom attributes and ISM Controls to a software policy, and creating multiple instances of Zip packages. See “Overview of Software Policies” on page 57 in this chapter for more information.

Library

In the SAS Client, the Library provides a way to display the feature (such as application configurations, software policies, patches, patch policies, packages, OS sequences, OS profiles) managed by Opware SAS. Folders are also located in the Library. In the Library, you can view the feature either by their type or by their location in the folder hierarchy. Some of the features such as patch policies, audit and remediation, and OS installation profiles, are not contained in folders, and they can be only viewed in the By Type view. The following table lists the feature objects displayed in the Library and whether they can be added to folders.

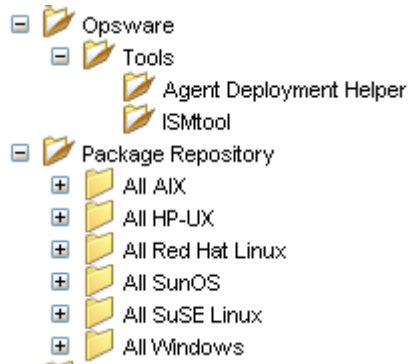
Table 2-1: Feature Objects displayed in the Library in the SAS Client

FEATURE OBJECTS	BY TYPE VIEW	BY FOLDER VIEW
Application Configurations	X	
Software Policies	X	X
Audit and Remediation	X	
Patches	X	
Patch Policies	X	
OS Installation Profiles	X	
OS Sequences	X	X
Packages	X	X

See the *Opware® SAS User's Guide: Application Automation* for more information about Audit and Remediation, Application Configuration, OS Provisioning, and Patch Management.

When you install Opsware SAS, the Library contains the following default folders:

Figure 2-2:



- The Opsware folder contains the tools required to install the Windows Agent Deployment Helper and upload ISMs to Opsware SAS.

See the *Opsware[®] SAS Planning and Installation Guide* for more information about Windows Agent Deployment Helper. See the *Opsware[®] SAS Content Utilities Guide* for more information about ISMs.

- The Package Repository folder contains packages that are organized by operating system families. See “Overview of Package Management” on page 93 in this chapter for information about packages.
- In addition to using the default folders, you can create new folders in the Library to manage the software in your IT environment. See “Creating a Folder” on page 90 in this chapter for information about how to create folders.

Overview of Software Policies

In Opsware SAS, software policies allow you to install software and configure applications simultaneously. A software policy can contain packages, RPM packages, patches, application configurations, and other software policies. After creating a software policy, you can attach it to servers or groups of servers. When you remediate a server or group of servers, the patches, packages, RPM packages, and application configurations specified in the attached policy are automatically installed and applied respectively. The remediation process works by comparing what is actually installed on a server to the software that should be installed on the server according to the server's model. Opsware

SAS then determines what operations are required to make the server conform to its model. See the *Opsware® SAS User's Guide: Application Automation* for more information about the remediation process.

When a software policy is attached to servers or groups of servers, it has a persistent association to those servers. As a result, whenever the software policy is updated, you receive a notification indicating which servers or groups of servers are affected by the updated software policy. You can then choose to remediate the servers or groups of servers to reflect the changes to the software policy. See the *Opsware® SAS User's Guide: Application Automation* for more information about attaching software policies to servers.

The Software Template provides you with an option of installing software and configuring applications simultaneously without persistently associating the software policy with servers or groups of servers. As a result, if the software policy is updated, the changes are not reflected on those servers. See the *Opsware® SAS User's Guide: Application Automation* for more information about the software template.

A software policy is associated with an operating system family. When you add software resources to a software policy, the software resources must belong to the same operating system family as the software policy. For example, if you define the operating system for a software policy as HP-UX, you can only add software resources applicable to versions of HP-UX (as supported by Opsware) to the software policy.

Similarly, if the operating system defined for a software policy is Windows 2000 and Windows 2003, the software resources that are applicable to Windows 2000 and Windows 2003 operating systems can be added to the software policy.

Software Policy Inclusion

A software policy can include other software policies. The included software policies and the parent software policy must belong to the same operating system family.

After you add the software resources to a software policy, you can specify the order in which you want them to be installed. When you attach a software policy to a server and remediate the server, Opsware SAS installs the software resources in the software policy the specified order.

When a software policy contains included software policies, all the software resources from the included software policies are grouped together and then installed as a unit.

In Opware SAS, policy inclusion provides a way to organize your software and manage dependencies between the software resources across included policies. You can achieve this by specifying the install order among packages, patches, and application configurations from an included policy. When you specify the install order from included policies, all the packages, patches, and application configurations in the included policy are grouped together and installed as a group.

You cannot specify the installation order for RPM packages in a software policy.



You must have permissions to set the installation order of packages and patches in a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Figure 2-3: Installation Order in a Software Policy

	Name	Location	Comes From
1	OPSWpytwist-32f.0.1.10.69.zip (Windows ...	/Opware/Tools/Pytho...	-
2	OPSWpytwist-32f.0.1.10.69.zip (Windows ...	/Opware/Tools/Pytho...	-
3	OPSWpytwist-32f.0.1.10.69.zip (Windows ...	/Opware/Tools/Pytho...	-
4	OPSWpytwist-32f.0.1.10.69.zip (Windows ...	/Opware/Tools/Pytho...	-
5	OPSWpytwist-32f.0.1.10.69.zip (Windows ...	/Opware/Tools/Pytho...	-
6	OPSWpytwist-32f.0.1.8.8.zip (Windows X...	/Opware/Tools/Pytho...	Windows Python UAPI Acc...
7	OPSWpytwist-32f.0.1.8.8.zip (Windows 20...	/Opware/Tools/Pytho...	Windows Python UAPI Acc...
8	OPSWpytwist-32f.0.1.8.8.zip (Windows 20...	/Opware/Tools/Pytho...	Windows Python UAPI Acc...
9	OPSWpytwist-32f.0.1.8.8.zip (Windows N...	/Opware/Tools/Pytho...	Windows Python UAPI Acc...
10	OPSWpytwist-32f.0.1.8.8.zip (Windows 20...	/Opware/Tools/Pytho...	Windows Python UAPI Acc...
11	OPSWpytwist-32f.0.1.10.2.zip (Windows 2...	/Opware/Tools/Pytho...	-
12			

In Figure 2-3, the software policy contains 11 packages. Packages 6 through 10 are inherited from the included software policy Windows Python UAPI. The following installation order is specified for the software policy:

1. Package 1 through 5 from the software policy
2. All packages and patches from the Software policy Windows Python UAPI
3. Package 11 from the software policy

When you specify the installation order, all the packages from the included software policy Windows Python UAPI are installed as a group. In the Policy Items window, the packages from an included software policy are indicated by a gray color.

See “Specifying the Installation Order in a Software Policy” on page 76 in this chapter for more information.



If a Software Template contains sub policies then during remediation, Opware does not consider the install order for the sub policies. For the software resources in a sub policy you can specify the install order and during remediation, all the software resources contained in the sub policies are then installed as per the install order.

Setting Custom Attributes For Software Policies

The SAS Client provides a data management function that allows you to set custom attributes for servers by using software policies. The custom attributes include miscellaneous parameters and named data values. You can write scripts that use these parameters and data values when you perform a variety of functions, including network and server configuration, notifications, and CRON script configurations.

Using the SAS Client, you can set custom attributes either for software policies or for servers or groups of servers directly. When you set a custom attribute for a software policy, the custom attributes and values affect all the servers attached to the policy. When a software policy containing other software policies is attached to a server, all the custom attributes and values from the parent software policy and the included software policies are added to the server.

Setting custom attributes to servers or groups of servers directly allows you to override the attributes and values set by a software policy. For example, if a certain port is required for installing an application, you can set it as a custom attribute in a software policy. When you attach the software policy to multiple servers, the attribute is added to those servers. If required, you can change the port settings of a particular server attached to the software policy, without changing the port settings of all the other servers attached to the software policy. You can achieve this by setting the custom attribute on the server directly. As a result, the custom attribute value set on the server directly supersedes the value set by the software policy for that server.

See “Adding Custom Attributes to a Software Policy” on page 82 in this chapter for more information.

Including ISM Controls in Software Policies

An Intelligent Software Module (ISM) is an installable software package created with the Opsware ISM Development Kit (IDK). An ISM can contain control scripts that perform day-to-day, application-specific tasks such as starting software servers. For example, an ISM for Apache might contain control scripts that start and stop the HTTP server.

In Opsware SAS you can create a control script with a text editor, package the script into an ISM, and then upload the ISM to Opsware SAS. See the *Opsware® SAS Content Utilities Guide* for more information about ISM control scripts. After you upload an ISM package into Opsware SAS by using the ISM tool in the IDK, the ISM appears in the SAS Client as a package in the Library. Using the SAS Client, you add the ISM package to a software policy and then attach the software policy to managed servers. See “Adding a Package to a Software Policy” on page 68 in this chapter for more information.

You can run the control scripts in the ISM with the Run ISM Control window of the SAS Client. See the *Opsware® SAS User's Guide: Application Automation* for information about running ISM Controls.

An ISM control script can have parameters corresponding to custom attributes. The name of a parameter matches the name of its corresponding custom attribute. The value of a custom attribute determines the value of the parameter. The source of a custom attribute is an Opsware SAS object, such as a facility, customer, server, group of servers, or software policy. Custom attributes with the same name (but with different values) can be specified on different Opsware SAS objects. If a server is associated with objects that have identically named custom attributes, Opsware SAS uses a predefined search order to determine the custom attribute that provides the parameter value. In the Run ISM Control window of the SAS Client, you can view the name and value of the control parameter.

See the *Opsware® SAS Content Utilities Guide* for more information on the search order for custom attributes.

Software Policies for Patch Installation

With Opsware SAS you can install patches on servers in the following ways:

- Using patch policies to install Windows patches. See the *Opsware® SAS User's Guide: Application Automation* for more information about Windows patch management.
- Using software policies to install Unix and Windows patches. See the *Opsware® SAS User's Guide: Application Automation* for more information about Unix patch management.

- Installing patches directly on servers. See the *Opware® SAS User's Guide: Application Automation* for more information about installing patches directly.



A software policy can contain both Unix patches and Windows patches. Opware Inc. recommends that you use patch policies to install Windows patches and software policies to install Unix patches on servers. You cannot install Unix patches by using patch policies.

Patch policies provide you with an option of setting a policy exception. If you need to include or exclude a Windows patch in a patch policy from being installed, you can deviate from a patch policy by specifying that Windows patch in a policy exception. You can also set precedence rules for applying patch policies and policy exceptions. The precedence rules determine the Windows patches that are actually installed on a server. See the *Opware® SAS User's Guide: Application Automation* for more information about precedence rules for applying patch policies and patch policy exceptions.

After you attach a patch policy to a managed server, the remediation process installs the Windows patches in a patch policy on the managed server. If you remove any patches from the patch policy and remediate the server again, the remediation process does not remove the Windows patches from the server.

However, with a software policy, the remediation process removes the patches from the server. There are some patches like Service Packs that cannot be uninstalled. For example, if you remove a Service Pack from a software policy and remediate the server again, the Service Pack is not uninstalled from the server.

Creating a Software Policy

A software policy contains software resources such as packages, patches, RPM packages, and application configurations that need to be installed on managed servers. In the SAS Client, you can create a software policy from either the By Type or the By Folder view in the Library.



You must have a set of permissions to create and manage a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Creating a Software Policy from the By Type View in the Library

Perform the following steps to create a software policy in the SAS Client:

- 1 From the Navigation pane, select **Library** ► **By Type** ► **Software Policies**. The list of software policies appears in the Content pane as shown in Figure 2-4. By default, the software policies are organized by operating system families.

Figure 2-4: Software Policies in the SAS Client

Name	Number of Objects
AIX	11
HP-UX	15
Linux	116
Red Hat	112
Solaris	12
SUSE	4
Windows	257

- 2 Select a specific operating system.
- 3 From the **Actions** menu, select **New**. The Software Policy window appears.
- 4 In the Name field, enter the name of the software policy.
- 5 To save changes, select **Save** from the **File** menu.
- 6 To set the properties for a software policy, see “Setting Software Policy Properties” on page 66.

Creating a Software Policy from the By Folder view in the Library

Perform the following steps to create a software policy in the SAS Client:


- 1** From the Navigation pane, select **Library ► By Folder**. The folder hierarchy in the Library appears in the Content pane.
- 2** Select the folder that should contain the software policy.
- 3** From the **Actions** menu, select **New Software Policy**. The Software Policy window appears.
- 4** In the Name field, enter the name of the software policy.
- 5** To save changes, select **Save** from the **File** menu.
- 6** To set the properties for a software policy see “Setting Software Policy Properties” on page 66.

Ways to Open a Software Policy

In the SAS Client, there are several ways to open a software policy. You can open a software policy from:

- The Search option in the Navigation pane
- The Devices option in the Navigation pane
- The By Type view in the Library
- The By Folder view in the Library
- The Compliance Dashboard

Opening a Software Policy from Search

- 1** From the Navigation pane, select **Search**.
- 2** Select Software Policy from the drop-down list and then enter the name of the policy in the text field.
- 3** Select . The search results appear in the Content pane.
- 4** From the Content pane, select the software policy and then select **Open** from the **Actions** menu. The Software Policy window appears.

Opening a Software Policy from Devices

- 1** From the Navigation pane, select **Devices** ► **All Managed Servers**. The server list appears in the Content pane.

Or

From the Navigation pane, select **Devices** ► **Device Groups**. The device groups list appears in the Content pane.

- 2** From the Content pane, select a server and then from the **Actions** menu, select **Open**. The Server Explorer window opens.
- 3** From the Views pane, select Software Polices. The software policies attached to the server appear in the Content pane.
- 4** From the Content pane, select the software policy and then select **Open** from the **Actions** menu. The Software Policy window appears.

Opening a Software Policy from the By Type view in the Library

- 1** From the Navigation pane, select **Library** ► **By Type** ► **Software Policies**. The software policies appear in the Content pane.
- 2** From the Content pane, select the software policy and then select **Open** from the **Actions** menu. The Software Policy window appears.

Opening a Software Policy from the By Folder view in the Library

- 1** From the Navigation pane, select **Library** ► **By Folder**. The folder hierarchy in the Library appears in the Content pane.
- 2** From the Content pane, select the software policy in a folder and then select **Open** from the **Actions** menu. The Software Policy window appears.

Opening a Software Policy from the Compliance Dashboard

- 1** From the Navigation pane, select **Library** ► **By Type** ► **Software Policies**. The software policies appear in the Content pane.
- 2** From the Content pane, select the software policy and then select **Open** from the **Actions** menu. The Software Policy window appears.

Setting Software Policy Properties

After you create a software policy, you can view and modify its properties. You can view properties such as the Opware user who created the software policy, the date when it was created, and the Opware ID of the software policy. You can also modify the name, description, availability, the location of the software policy in the Library and the operating systems of the software policy.



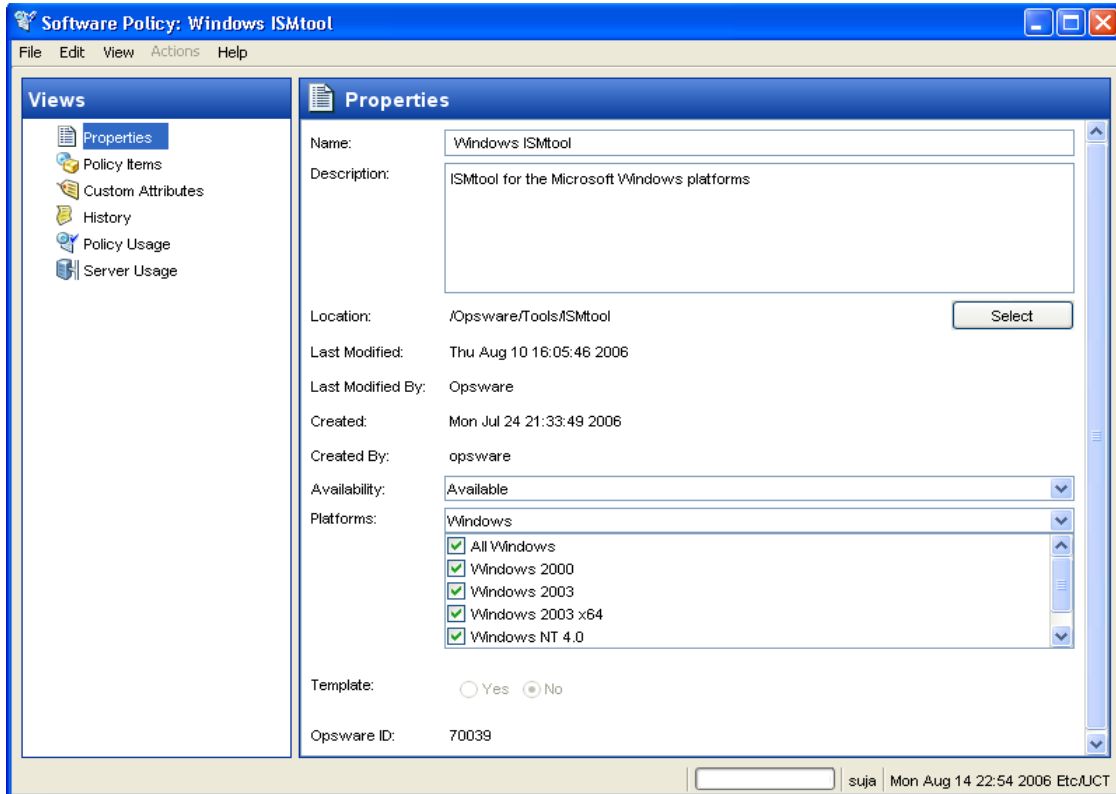
You must have a set of permissions to manage software policies. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to define the properties of a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.

- 2 From the Content pane, select the software policy and open it. The Software Policy window appears as shown in Figure 2-5.



Figure 2-5: Software Policy Window



- 3 From the Views pane, select Properties. You can specify the name, description, location, life cycle, and operating systems for the software policy in the Content pane.
- 4 In the Name field, enter a name for the software policy.
- 5 In the Description field, enter text that describes the purpose or contents of the policy.
- 6 Click **Select** to specify the location for the software policy in the folder hierarchy. The Select Location window appears. Select a folder in the Library to specify the location of the software policy and then click **Select**.
- 7 From the Availability drop-down list, select the Opsware server life cycle values for the software policy.

- 8** From the Platforms drop-down list, select the operating system family or specific operating systems in that family.
- 9** In the Template field, select Yes to designate a software policy as a template. A software policy template is not persistently associated with a server. See the *Opware® SAS User's Guide: Application Automation* for information about installing software policy template.
- 10** To save the changes, select **Save** from the **File** menu.



In the SAS Client, a software policy is represented by the icon . A software template is represented by the icon .

Adding a Package to a Software Policy

After you create a software policy, you can add packages to it. When you add packages to a software policy, the packages must belong to the same operating system family as the software policy. Adding packages to a software policy does not install the packages on a managed server. After you add packages to a software policy, you must attach it to a managed server and then remediate the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



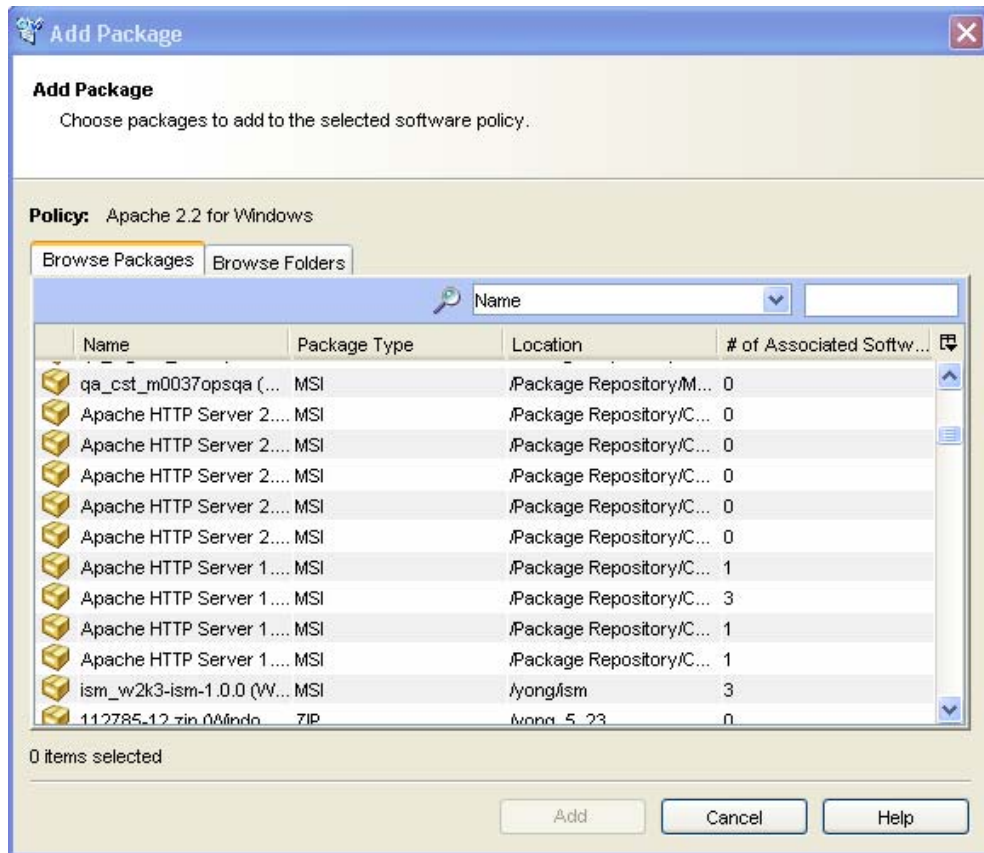
You must have a set of permissions to add packages to a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to add packages to a software policy:

- 1** From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Packages and Patches.

- 4** From the **Actions** menu, select **Add Packages**. The Add Package window appears as shown in Figure 2-6.

Figure 2-6: Add Package Window in the SAS Client



- 5** Select Browse Packages to display a list of packages that can be added to the software policy. The package list includes the following information about the package:
- The name of the package
 - The package type
 - The location of the package in the folder hierarchy
 - The date when the package was last modified
 - The Opware user who last modified the package
 - The size of the package

- The number of software policies associated with the package
- The number of server which have the package installed
- The Opware SAS unique ID for the package
- The platform associated with the package

Or

Select Browse Folders to display the folder hierarchy in the Library and the list of packages contained in the folders. The package list includes the following information:

- The name of the package
- The package type
- The date when the package was last modified
- The Opware user who last modified the package
- The date when the package was created
- The Opware user who created the package

6 Select the packages and click **Add**. The selected packages appear in the Content pane.

7 To save the changes, select **Save** from the **File** menu.

Adding a Patch to a Software Policy

After you create a software policy, you can add patches to it. When you add patches to a software policy, the patches must belong to the same operating system family as the software policy. Adding patches to a software policy does not install the patches on a managed server. After you add packages to a software policy, you must attach it to a managed server and then remediate the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to add patches to a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to add patches to a software policy:

1 From the Navigation pane, select **Library > By Type > Software Policies**.

- 2 From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3 From the Views pane, select Policy Items and then Patches.
- 4 From the **Actions** menu, select **Add Patches**. The Add Patches window appears.
- 5 Select patches from the patches list to be added to the policy. The patch information in the patch list is different for Unix patches and Windows patches. The Unix patch list includes the following information about the patch:
 - The name of the patch
 - The patch type
 - The status of the patch within Opsware SAS
 - The description of the patch
 - The Opsware SAS unique ID for the patchThe Windows patch list includes the following information about the patch:
 - The name of the patch
 - The patch type
 - The Microsoft severity ratings for the patch
 - The date when Microsoft released the patch
 - The status of the patch within Opsware SAS
 - The Microsoft Security Bulletin ID for the patch
 - The Microsoft Knowledge Base article ID number for the patch
 - The description of the patch
 - The Opsware SAS unique ID for the patch
 - The patch locale
- 6 Select the patches and click **Add**. The selected patches appear in the Content pane.
- 7 To save the changes, select **Save** from the **File** menu.

Adding an RPM Package to a Software Policy

After you create a software policy, you can add RPM packages to the software policy. When you add RPM packages to a software policy, the packages must belong to the same operating system family as the software policy. Adding RPM packages to a software policy does not install the RPM packages on a managed server. After you add the RPM

packages to a software policy, you must attach the software policy to a managed server and then remediate the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to add RPM packages to a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to add RPM packages to a software policy:

- 1** From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then RPM Packages.
- 4** From the **Actions** menu, select **Add RPM Package**. The Add RPM Package window appears.
- 5** Select Browse Packages to display a list of RPM packages that can be added to the software policy. The package list includes the following information about the package:
 - The name of the RPM package
 - The RPM Package type
 - The location of the RPM package in the folder hierarchy
 - The date when the RPM package was last modified
 - The Opware user who last modified the RPM package
 - The size of the RPM package
 - The number of software policies associated with the RPM package
 - The number of server which have the RPM package installed
 - The Opware SAS unique ID for the RPM package
 - The platform associated with the RPM package

Or

Select Browse Folders to display the folder hierarchy in the Library and the list of RPM packages contained in the folders. The RPM package list includes the following information:

- The name of the RPM package
- The RPM Package type
- The date when the RPM package was last modified
- The Opware user who last modified the RPM package
- The date when the RPM package was created
- The Opware user who created the RPM package

6 Select the packages and click **Add**. The selected RPM packages appear in the Content pane.

7 To save the changes, select **Save** from the **File** menu.

Adding an Application Configuration to a Software Policy

After you create a software policy, you can add application configurations to it. When you add application configurations to a software policy, the application configuration must belong to the same operating system family as the software policy. Adding application configurations to a software policy does not change the configurations on a managed server. After you add application configurations to a software policy, you must attach the software policy to a managed server and then remediate the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to add application configurations to a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to add application configurations to a software policy:

- 1** From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Application Configurations.

- 4** From the **Actions** menu, select **Add Application Configuration**. The Add Application Configuration window appears. This window contains the list of application configurations that can be added to the software policy. The application configuration list includes the following information:
 - The name of the application configuration
 - The operating systems associated with the application configuration
 - The version number of the application configuration
 - The customers associated with the application configuration
 - The number of configuration files contained in the application configuration
 - The date the application configuration was last modified
 - The Opware user who last modified the application configuration
 - The description of the application configuration
- 5** Select the application configurations and click **Add**. The selected application configurations appear in the Content pane.
- 6** To save the changes, select **Save** from the **File** menu.

Adding a Software Policy to a Software Policy

After you create a software policy, you can add other software policies to it. When you add software policies to a software policy, the software policies must belong to the same operating system family as the parent software policy. Adding software policies to a software policy does not install the software resources on a managed server. After you add software policies to a software policy, you must attach it to a managed server and then remediate the software policy.



You must have a set of permissions to add software policies to a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to add software policies to a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.

- 3** From the Views pane, select Policy Items and then Software Policies.
- 4** From the **Actions** menu, select **Add Software Policy**. The Add Software Policy window appears.
- 5** Select Browse Software Policies to display a list of software policies that can be added to the software policy. The software policy list includes the following information:
 - The name of the software policy
 - The location of the software policy in the folder hierarchy
 - The date when the software policy was last modified
 - The Opware user who last modified the software policy
 - The Opware user who created the software policy
 - The date when the software policy was created
 - The number of packages in the software policy
 - The number of patches in the software policy
 - The number of application configurations in the software policy
 - The number of software policies included in the software policy
 - The number of software policies associated with the software policy
 - The number of servers which have the software policy attached
 - The description of the software policy
 - The Opware SAS unique ID for the software policy

Or

Select Browse Folders to display the folder hierarchy in the Library and the list of software policies contained in the folders. The software policy list includes the following information:

- The name of the software policy
- The type of software
- The Opware user who last modified the software policy
- The date when the software policy was last modified
- The Opware user who created the software policy
- The date when the software policy was created

- 6** Select the software policies and click **Add**. The selected software policies appear in the Content pane.
- 7** To save the changes, select **Save** from the **File** menu.

Specifying the Installation Order in a Software Policy

Once you have added the software resources to a software policy, you can specify the installation order among packages, patches, and application configurations in the software policy. When you specify the installation order for the included policies, all the packages, patches, and application configurations in the included policy are grouped together and installed as a unit.



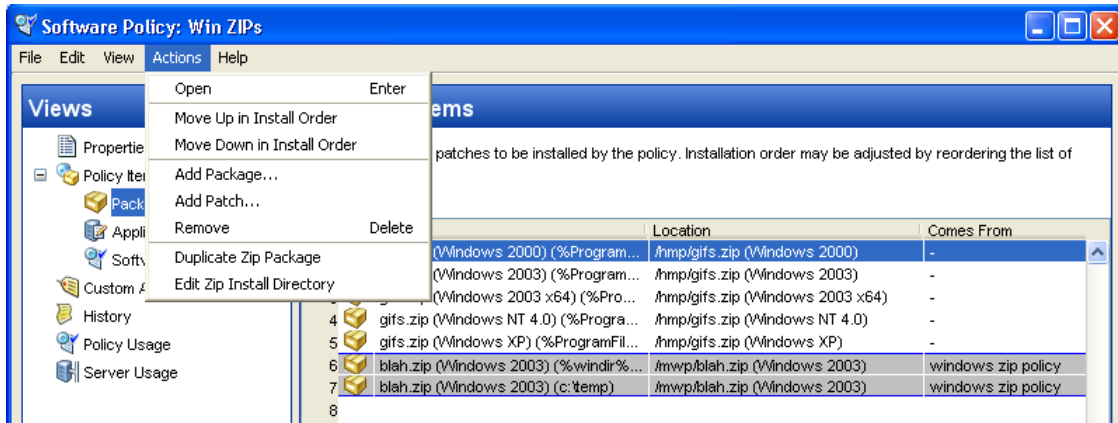
You must have a set of permissions to specify the installation order of the software resources in a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to specify the installation order in a software policy:

- 1** From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Packages and Patches.
- 4** In the Content pane, the list of all packages and patches contained in the software policy appears. The list contains the following information:
 - The name of the package or patch
 - The location of package or patch in the folder hierarchy
 - The software policy from which the package or patch comes from

- To specify the installation order, select the packages and patches, and then from **Actions** menu, select **Move Up in Install Order** or **Move Down in Install Order**.

Figure 2-7: Specifying Installation Order



- To save the changes, select **Save** from the **File** menu.

Setting Installation and Update Options for an RPM Package

Once you have added an RPM package to a software policy, you can specify if the RPM package needs to be installed on the server or upgraded to the latest version during remediation. To install or upgrade the RPMs on a managed server, you must remediate the server with the software policy.

In a software policy, you can specify the installation and update options for RPM packages only if they are directly added to the software policy. You cannot specify the installation and upgrade options for RPM packages if they are included in this software policy.



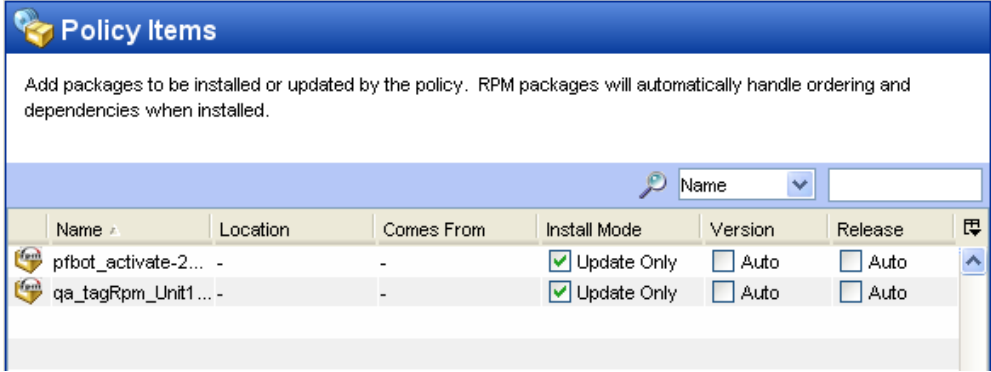
You must have a set of permissions to specify the installation options of the RPM packages in a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to specify the installation options for an RPM package:

- From the Navigation pane, select **Library > By Type > Software Policies**.
- From the Content pane, select the software policy and open it. The Software Policy window appears.

- 3 From the Views pane, select Policy Items.
- 4 From the Views pane, select RPM Packages. The list of all RPM packages contained in the software policy appears in the Content pane as shown in the figure below.

Figure 2-8: Setting Installation and Upgrade Options for RPM Packages



The screenshot shows a window titled "Policy Items" with a search bar and a table of RPM packages. The table has the following columns: Name, Location, Comes From, Install Mode, Version, and Release. Two packages are listed: "pfboot_activate-2..." and "qa_tagRpm_Unit1...". Both have "Update Only" checked in the Install Mode column and "Auto" checked in the Version and Release columns.

Name	Location	Comes From	Install Mode	Version	Release
pfboot_activate-2...	-	-	<input checked="" type="checkbox"/> Update Only	<input type="checkbox"/> Auto	<input type="checkbox"/> Auto
qa_tagRpm_Unit1...	-	-	<input checked="" type="checkbox"/> Update Only	<input type="checkbox"/> Auto	<input type="checkbox"/> Auto

- 5 Select an RPM package. For every RPM package, you can specify the following options:
 - In the Install Mode column, select the Update Only check box to update the RPM version on the managed server to the version specified in the software policy.
 - In the Install Mode column, clear the Update Only check box to install the RPM packages specified in the software policy on the managed server.
 - In the Version column, select the Auto check box to automatically update the RPM package in the software policy to a newer version or release of the RPM.



Opware SAS will update the RPM package in the software policy to a version or release of the RPM only if the newer release or version is placed in the same folder as the RPM package specified in the software policy.

- In the Release column, select the Auto check box to automatically update the RPM package in the software policy to a newer release of the same version of the RPM.



Opware SAS will update the RPM package in the software policy to a newer release of the RPM only if the newer release of the same version is placed in the same folder as the RPM package specified in the software policy.

See “Install and Update of RPM Packages Using a Software Policy” on page 120 in this chapter for more information.

6 To save the changes, select **Save** from the **File** menu.

Removing a Package from a Software Policy

Removing a package from a software policy does not uninstall the package from a managed server. It only removes the package from the software policy. To uninstall the packages from a managed server, you must remediate the software policy. See the *Opware[®] SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to remove packages from a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware[®] SAS Administration Guide* for more information.

Perform the following steps to remove a package from a software policy:

- 1** From the Navigation pane, select **Library** ► **By Type** ► **Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Packages and Patches.
- 4** Select the packages that you want to remove from the package list displayed in the Content pane.
- 5** From the **Actions** menu, select **Remove**.
- 6** To save the changes, select **Save** from the **File** menu.

Removing a Patch from a Software Policy

Removing a patch from a software policy does not uninstall the patch from a managed server. It only removes the patch from the software policy. To uninstall the patch from a managed server, you must remediate the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to remove a patch from a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to remove a patch from a software policy:

- 1** From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Packages and Patches.
- 4** Select the patches that you want to remove from the patch list displayed in the Content pane.
- 5** From the **Actions** menu, select **Remove**.
- 6** To save the changes, select **Save** from the **File** menu.

Removing an RPM Package From a Software Policy

Removing an RPM package from a software policy does not uninstall the package from a managed server. It only removes the RPM package from the software policy. To uninstall the RPM packages from a managed server, you must detach the software policy from the server and then remediate the server against the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to remove the RPM packages from a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to remove an RPM package from a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then RPM Packages.
- 4** Select the RPM packages that you want to remove from the package list.
- 5** From the **Actions** menu, select **Remove**.

To save the changes, select **Save** from the **File** menu.

Removing an Application Configuration from a Software Policy

Removing an application configuration from a software policy does not remove the configuration from a managed server. It only removes the application configuration from the software policy. To reconfigure the managed server, you must remediate the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to remove an application configuration from a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to remove an application configuration from a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Application Configurations.
- 4** Select the application configurations that you want to remove from the list displayed in the Content pane.
- 5** From the **Actions** menu, select **Remove**.
- 6** To save the changes, select **Save** from the **File** menu.

Removing a Software Policy from a Software Policy

Removing a software policy from a software policy does not remove the software in the policy from a managed server. It only removes the software policy from the software policy. To remove the software from a managed server, you must remediate the software policy. See the *Opware® SAS User's Guide: Application Automation* for more information about the remediation process.



You must have a set of permissions to remove a software policy from a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to remove a software policy from a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Software Policies.
- 4** Select the software policies that you want to remove from the list displayed in the Content pane.
- 5** From the **Actions** menu, select **Remove**.
- 6** To save the changes, select **Save** from the **File** menu.


Adding Custom Attributes to a Software Policy

When you add a custom attribute to a Software Policy, the attribute values affect the servers attached to the software policy. After you add a custom attribute to a software policy, you must attach it to a managed server and then remediate the software policy.



You must have a set of permissions to add custom attributes to a software policy. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to add a custom attribute to a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Custom Attributes.
- 4** Click **Add**.
- 5** In the Name field, enter the name of the custom attribute.
- 6** In the Value field click . The Input dialog appears. Enter the value for the custom attribute.
- 7** To save the changes, select **Save** from the **File** menu.

Editing Custom Attributes in a Software Policy

Perform the following steps to edit a custom attribute:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Custom Attributes.
- 4** Select the custom attribute that you want to edit.
- 5** Update the name and value for the custom attribute in the Content pane.
- 6** To save the changes, select **Save** from the **File** menu.

Deleting Custom Attributes from a Software Policy


Perform the following steps to delete a custom attribute:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Custom Attributes.
- 4** From the Content pane, select the custom attribute that you want to delete and then click **Remove**.
- 5** To save the changes, select **Save** from the **File** menu.

Adding Custom Attributes to Servers

Using the SAS Client, you can assign custom attributes to servers or groups of servers directly. This allows you to override the custom attribute set by a software policy.

Perform the following steps to add a custom attribute to a managed server:

- 1** From the Navigation pane, select **Devices ► All Managed Servers**.
- 2** From the Content pane, select the server and open it. The Server Explorer window appears.
- 3** From the Views pane, select Custom Attributes.
- 4** Click **Add**.
- 5** In the Name field, enter the name of the custom attribute.
- 6** In the Value field click . The Input dialog appears. Enter the value for the custom attribute.
- 7** To save the changes, select **Save** from the **File** menu.

Duplicating Zip Packages

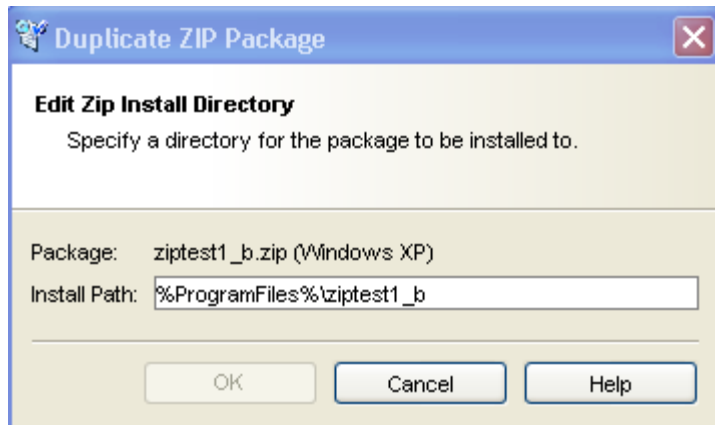
The Software Management feature allows you to install multiple instances of an application on a single server by using ZIP packages in a software policy. (In Opware SAS, these ZIP packages are sometimes referred to as “relocatable ZIPs.”) You can install the same ZIP package with different installation paths in multiple locations on a single server. Opware SAS supports installation of ZIP packages on both Unix and Windows operating systems. If the ZIP package was created with the IDK, then you cannot install the ZIP package into multiple locations on a single server.

Perform the following steps to create ZIP packages with different installation paths:

- 1** From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2** From the Content pane, select the software policy containing the ZIP package and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Items and then Packages and Patches.
- 4** From the Content pane, select the ZIP package.

- 5 From the **Actions** menu, select **Duplicate Zip Package**. The Duplicate ZIP Package window appears as shown below.

Figure 2-9: Duplicate ZIP Package Window in the SAS Client



- 6 In the Install Path field, enter the path where you will install the ZIP file. If you do not enter a path, the default directory for the Windows ZIP package is:

```
%SystemDrive%\Program Files\[basename of zip file]
```

The default directory for Unix Zip is:

```
/usr/local/[basename of zip file]
```

- 7 Click **OK** to install the Zip file.

Editing the ZIP Installation Directory

Perform the followings steps to change the default installation directory for ZIP packages:

- 1 From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2 From the Content pane, select the software policy containing the ZIP package and open it. The Software Policy window appears.
- 3 From the Views pane, select Policy Items and then Packages and Patches.
- 4 From the Content pane, select the ZIP package.
- 5 From the **Actions** menu, select **Edit Zip Install Directory**. The Edit ZIP Install Directory window appears.
- 6 In the Install Path field, enter the new path. If you do not enter a path, the default directory is for Windows ZIP package is:

%SystemDrive%\Program Files\[basename of zip file]

The default directory for Unix ZIP package is:

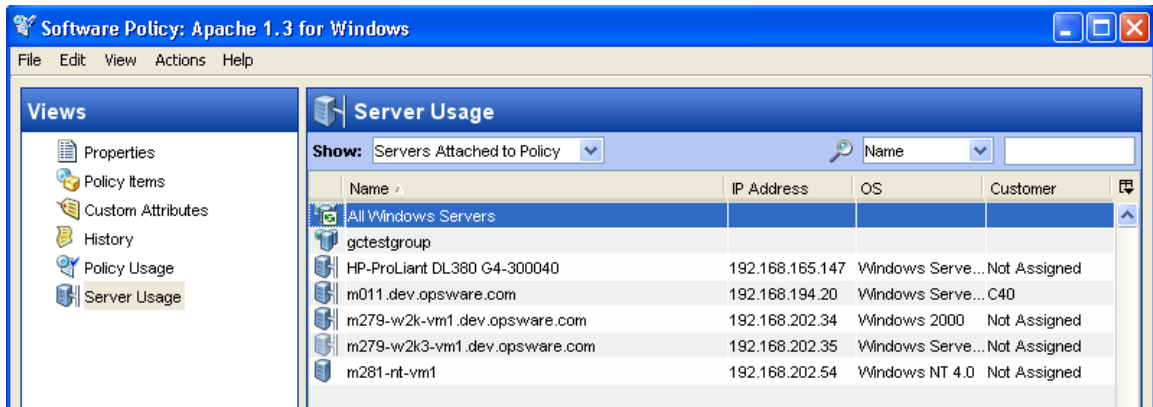
\usr\local\[basename of zip file]

- 7** Click **OK** to change the default installation directory for ZIP packages.

Viewing Servers Attached to a Software Policy

In the SAS Client, you can view the list of all servers attached to a Software Policy and servers that are detached from a software policy and not yet remediated from the software policy. In the Software Policy window, the servers that are detached from a software policy and not yet remediated from the software policy are represented by a gray icon as shown in Figure 2-10.

Figure 2-10: Server Usage in the Software Policy Window



Perform the following steps to view the servers attached to a software policy:

- 1** From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Server Usage. Select Servers Attached to Policies from the Show drop-down list. The list of servers attached to the software policy appears in the Content pane.
- 4** (Optional) Use the Show drop-down list to display the compliance information for a server with respect to a software policy.

Viewing All the Software Policies Associated with a Software Policy

A software policy can contain other software policies. In the Software Policy window, you can view all the software policies that contain the selected policy.

Perform the following steps to view software policies associated with a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select Policy Usage. The list of software policies associated with the selected software policy appears in the Content pane.

Viewing the History of a Software Policy

Perform the following steps to view the events associated with a software policy:

- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and open it. The Software Policy window appears.
- 3** From the Views pane, select History. The events associated with the software policy will display in the Content pane. You can view the action performed on a software policy, the user who performed the action, and the time when the action was performed.

Locating Software Policies in Folders

Perform the following steps to locate a software policy in the folder hierarchy:

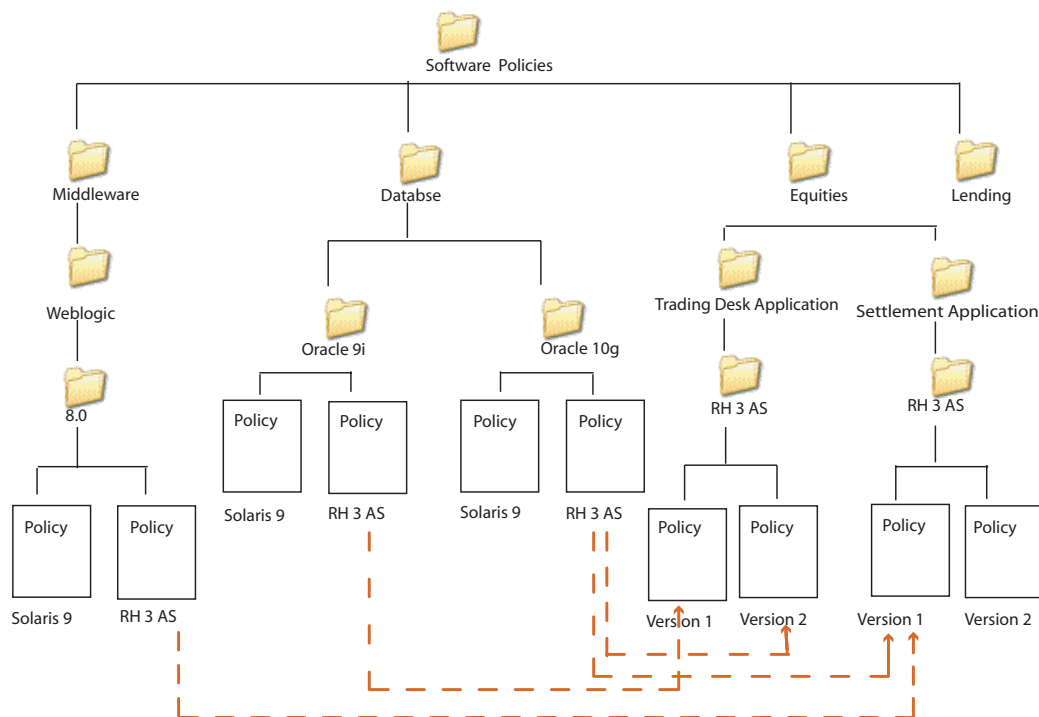
- 1** From the Navigation pane, select **Library > By Type > Software Policies**.
- 2** From the Content pane, select the software policy and then select **Locate in Folders** from the **Actions** menu. The folder hierarchy for the software policy appears in the Content pane.

Folders

The folder hierarchy in the Software Management feature provides a way to organize your software resources and allows you to define security permissions to control access to the contents of a folder. Folders can contain packages, software policies, and OS sequences. They can also contain other subfolders to form a hierarchal structure.

Folders act as containers that organize and manage your software resources to correspond to your IT environment. For example, you can organize the folders by functionality (Finance, Engineering, Operations, or Marketing), by applications (Web Servers, Web Application Servers, Database Servers, or Middleware), or by operating system versions (Unix, or Windows).

Figure 2-11: Example of a Hierarchical Folder Structure in the SAS Client



Opware SAS not only provides you with the flexibility of organizing folders based on functionality, applications, or OS versions, but it also allows you to share content among user groups. In this example, all software policies related to database servers are organized in the Database folder. The Database folder contains subfolders for Oracle 9i and Oracle 10g, which contain software policies for Oracle 9i and Oracle 10g, respectively. In the Oracle 9i and Oracle 10g folders, the software policies are organized based on the operating system versions. The Oracle 10g folder contains software policies for various operating systems such as Solaris 9 and Red Hat 3 AS.

Similarly, the Settlement Application subfolder contains the software policies necessary for that functionality. The software policies in this folder are organized based on operating system versions. The Version 1 software policy in the Red Hat 3 AS subfolder references the policy for Oracle 10g (Red Hat 3 AS version) and the policy for Weblogic 8.0 (Red Hat 3.0 AS version).

When you attach the Version 1 software policy to a server, all the software in software policy Version 1 is installed in addition to the software in software policy Oracle 10g (Red Hat 3 AS version) and Weblogic 8.0 (Red Hat 3.0 AS version).

Folders cannot be attached to a server directly. Instead, you have to add the software resources in folders to a software policy and attach the software policy to a server. Folders also do not support inheritance, which means that the subfolders do not inherit the software resources of a parent folder. See “Creating a Folder” on page 90 in this chapter for more information. See “Overview of Software Policies” on page 57 in this chapter for more information.

Folders and Permissions

Folders allow you to define security boundaries to control access to their content across user groups. You can assign permissions to folders to determine who can access the contents of the folder such as software policies, packages, and OS Sequences. A folder's permissions determine the user groups that can view, create, modify, and delete items within the folder. A folder's permissions apply only to the items directly under the folder. They do not apply to items lower down in the hierarchy, such as the subfolders and subfolders of subfolders (grandchildren).

In addition to the Folder permissions, a user must have the appropriate SAS Client Feature permissions to access the contents of a folder. SAS Client Feature permissions determine what actions users can perform with the SAS Client, whereas the Folder permissions specify which folders users have access to.

See the *Opware[®] SAS Administration Guide* for more information about Folder permissions.

You can assign the following permissions to a folder, by associating a user group with each folder:

- **List Contents of Folder:** Enables a user to navigate to the folder in the hierarchy, view the folder's properties, and view the names of the folder's children.
- **Read Objects Within Folder:** Enables a user to view and use the contents of a folder.

- **Write Objects Within Folder:** Enables a user to view, use, and modify the contents of a folder.
- **Edit Folder Permissions:** Enables a user to modify the folder permissions or add customers to a folder. This permission delegates the management of folder permissions to another user group.

See the *Opware® SAS Administration Guide* for more information about types of Folder permissions.

In addition to Folder permissions, you can assign customer constraints to folders. See the *Opware® SAS Administration Guide* for more information about customer constraints.

Creating a Folder



You must have a set of permissions to create and manage folders. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to create a folder in the SAS Client:

- 1** From the Navigation pane, select Library ► By Folders.
- 2** From the **Actions** menu, select **New Folder**.
The name of the folder that you just created is New Folder (n), where n is a number based on the number of new folders already in existence.
- 3** Enter the name of the folder in the Content pane.
- 4** From the Navigation pane, select **Save** to save a folder.



To create a folder in a specific location, navigate to the desired location in the folder hierarchy and select **New Folder** from the **Actions** menu.



You can also rename, move, cut, and copy folders by selecting **Rename**, **Move**, **Cut**, and **Copy** from the **Actions** menu.

Setting Folder Properties

After you create a folder, you can view and modify the properties of the folder. You can view the folder properties, such as the Opware user who created the folder, the date when the folder was created, the location of the folder in the Library, and the number of subfolders and feature objects present in the folder. You can also modify the name and the description of the folder.

You can also set the Folder permissions and Customer permissions on the folder. See the *Opware® SAS Administration Guide* for information about setting Folder and Customer permissions.



You must have a set of permissions to manage folders. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to manage the properties of a folder in the SAS Client:

- 1** From the Navigation pane, select Library ► By Folders.
All the folders in the Library appear in the Content pane.
- 2** From the Content pane, select a folder.

- From the **Actions** menu, select **Folder Properties**. The Folder Properties window appears as shown in Figure 2-12.

Figure 2-12: Setting Folder Properties in the SAS Client



- Select the General tab to view the folder properties, such as the location of the folder in the Library, the Opware ID associated with the folder, and the features and subfolders contained in the folder.
- In the Name field, modify the name of the folder. In the Description field, enter a description of the folder.
- Click **OK** to save the changes or click **Cancel** to close this window without saving the changes.

Deleting a Folder



To delete a folder, you must have the required permissions. To delete a folder containing subfolders, you must have the required permissions for the subfolders as well as the parent folder. To obtain these permissions, contact your Opsware administrator. See the *Opsware® SAS Administration Guide* for more information.

Perform the following steps to delete a folder in the SAS Client:

- 1** From the Navigation pane, select Library ► By Folders.
All the folders in the Library appear in the Content pane.
- 2** From the Content pane, select the folder that you want to delete.
- 3** From the **Actions** menu, select **Delete**. The Confirmation window appears.
- 4** Click **Delete** to delete the folder.

Overview of Package Management

Packages are made available in Opsware SAS by uploading the packages to the Software Repository with the SAS Client or by using the Opsware Command Line Interface (OCLI) Version 1.0.

The Software Repository provides a data store for all software that the SAS Client manages. After you upload packages to the Software Repository, you can install packages by adding packages to software policies, attaching software policies to servers, and then performing a server remediation.

Each operating system that Opware SAS supports has a list of package types that you can upload. Opware SAS supports these package types on the supported operating systems, as shown in Table 2-2.

Table 2-2: Supported Operating Systems and Package Types

OPERATING SYSTEM	PACKAGE TYPE	FILE FORMATS	ADDITIONAL METADATA*
AIX	LPP (contains an update fileset or base filesets)	.bff, .l, .U, .lpp	N/A
	RPM	.rpm	N/A
	ZIP	.zip	N/A
HP-UX	Depot (contains products and filesets)	.tar, .depot	N/A
	ZIP	.zip	N/A
Linux	RPM	.rpm	N/A
	ZIP	.zip	N/A
Solaris	Patch	.jar, .tar, tar.gz, .tar.Z, t.gz, .zip	N/A
	Patch Cluster (contains patches)	.tar, .tar.gz, tar.Z, .t.gz, .zip	N/A
	Solaris package (contains package instances)	.pkg, .tar	N/A
	RPM	.rpm	N/A
	ZIP	.zip	N/A

Table 2-2: Supported Operating Systems and Package Types (continued)

OPERATING SYSTEM	PACKAGE TYPE	FILE FORMATS	ADDITIONAL METADATA*
Windows	Hotfix	.exe	N/A
	Security Patch	.exe	N/A
	MSI	.msi	N/A
	OS Service Pack	.exe	Service Pack Level
	Windows Utility (Microsoft Security Baseline Analyzer and qchain)	.exe	N/A
	Microsoft Patch Database (contains a description of available patches) See "About the Microsoft Patch Database" on page 412 in Chapter 8 for more information.	.xml, .cab	N/A
	ZIP	.zip	N/A
OS Independent	Unknown	All	N/A

* For certain package types, Opsware SAS requires that you provide additional metadata for the package.

AIX Packages

LPPs are the container packages for AIX. LPPs have the following characteristics:

- An LPP contains either one or more base filesets or an update fileset.
- When an LPP contains multiple filesets, frequently only a subset of those filesets is installed because users might want to install only certain filesets.

The basic unit of AIX packages is the fileset. Filesets have the following characteristics:

- Filesets are versioned.
- The two types of filesets are base and update.
- Users add filesets to software policies. Therefore, Opsware SAS adds filesets to and removes filesets from servers through remediation.

Filesets are delivered as part of an LPP file, which users upload to the Software Repository. Opware SAS automatically creates package entries for all the filesets that the LPPs contain. When viewing an LPP in the SAS Client, users see which filesets it contains.

The Opware Agent reports which filesets and Authorized Program Analysis Reports (APARs) are installed on servers because servers only report filesets and APARs (and cannot report LPPs). The SAS Client shows filesets and APARs in the Installed Packages list for a server.

LPP Metadata

Opware SAS uses the metadata contained in LPPs when creating the package entries in the list of packages. An LPP contains the following metadata:

- The name of the LPP
- The name, version, and description of each fileset in the LPP
- For an updated fileset, a list of APARs addressed by the fileset
- For each APAR listed, the list of filesets that make up that APAR



Opware SAS does not support bundles (which are abstract sets of filesets, drawn from multiple LPPs) or Program Temporary Fix (PTFs), which are similar to APARs without the metadata. However, users can still model a bundle or PTF by creating a software policy and attaching the filesets included in the bundle or PTF to that software policy.

When a user uploads an LPP, Opware SAS performs the following actions:

- Opens the LPP and parses its metadata.
- Automatically creates entries in the list of packages for the filesets in the LPP and registers them as installable.
- Automatically creates entries in the list of packages for the APARs defined by the update filesets in the LPP (if any).
- Registers the LPP as a non-installable package.

HP-UX Packages

Depots are the container packages for HP-UX. Depots have the following characteristics:

- A depot either contains products that contain filesets, or it contains patch products that contain patch filesets.
- When a depot contains multiple products and filesets, frequently only a subset is installed because users might want to install only certain products or filesets.
- A depot is a special type of directory formatted for use by HP Software Distributor (SD-UX) commands. SD-UX, a software management system, is the distribution mechanism for all HP software for HP-UX.
- A depot can be a local directory, a CD-ROM, tape, or it can reside on a server on the network.
- Multiple depots can be created for different applications or purposes.
- Users upload depots to the Software Repository in TAR format.
- Users can upload depots as HP-UX 11.00 or 11.11 depots. However, HP-UX software can be compatible with both 11.00 and 11.11. When the software in a depot is compatible with both 11.00 and 11.11, upload the depot to the Software Repository for both 11.00 and 11.11.
- Depots cannot be differentiated by a hardware platform, such as s700 or s800.
- HP-UX depots have two basic formats:
 - **Directory:** The format for depots saved on a server or CD-ROM.
 - **Tape:** The format for standalone depot files and the format required for uploading HP-UX packages into Opware SAS.

Products and filesets are the installable packages for HP-UX. They have the following characteristics:

- Products and filesets are versioned.
- Filesets are the smallest installable unit. A fileset can belong to only one product, but can be included in multiple subproducts or bundles.
- Subproducts are logically related filesets and are not versioned; for example, X11.Manuals.
- Products are supersets of filesets.
- Bundles are logical groups of filesets; for example, HP-UX Support Tools Bundle.

Opware SAS supports products, filesets, and patch products as installable software.



Opware SAS does not support bundles (which are abstract sets of filesets, drawn from depots) or subproducts by automatically creating software policies for bundles and subproducts when users upload depots. However, users can still model bundles and subproducts by creating software policies for them and attaching the filesets for the bundles and subproducts. Opware SAS does not support using HP-UX codewords.

When a user uploads a depot, Opware SAS performs the following actions:

- Opens the depot and parses its metadata.
- Automatically creates entries in the list of packages for the products and filesets in the depot and registers them as installable.
- Registers the depot as a non-installable package.



If a depot contains different software for HP-UX 11.00 and 11.11, create OS-specific depots for each HP-UX version and upload the depots to the Software Repository. The SAS Client does not check the OS compatibility of the products and filesets in a depot when a user uploads the depot. When adding products or filesets to a software policy, the products and filesets can be added only when the associated OS of their depot matches the OS specified for the software policy.

The format of HP-UX version information can be inconsistent, making it difficult to determine whether one version is older than another when installing a package that has another version already installed. Opware SAS attempts to install it anyway. An error results if a newer version is already installed.



Opware SAS does not provide alternate root support for HP-UX. Do not include commands that require alternate root support in the Install Flags text box of the Packages: Properties page. By default, the HP-UX `swinstall` command does *not* replace a newer version of a fileset or product with an older version. However, Opware SAS does overwrite newer versions of filesets and products with older versions. Opware SAS does not support relocating packages for HP-UX.

Depot Metadata

Opware SAS uses the metadata contained in depots when creating the package entries in the list of packages. A depot contains the following metadata:

- The name, version, and description of each product in the depot
- The list of filesets in each product in the depot
- The name, version, and description of each fileset in the depot

Preparing for HP-UX Package Management

Before you upload a depot to the Software Repository, perform the following tasks:

- 1** Convert the depot on the installation media (CD-ROM) from directory format to tape format by using the `swpackage` command:

```
swpackage -x media_type=tape -s <directory depot> <software selection> @ <file depot>
```

- 2** Split the depot into depots for each product.



You can perform this step manually by using NIM utilities or you can run a script to automate this step. See “Example: File - Script to Split a Depot by Product” on page 99 in this chapter for more information. See “Example: File - Script to Split a Depot by Bundle” on page 100 in this chapter for more information.

Example: Commands - Converting a Depot

The following example shows the commands used to create a Quality Pack file depot from the Support Plus CD-ROM for HP-UX 11.00:

- 1** Mount the directory on the CD-ROM that contains the Quality Pack file depot:

```
mount -F cdfs /dev/dsk/c2t1d0 /cdrom
```
- 2** Convert the depot on the CD-ROM from directory format to tape format by using the `swpackage` command:

```
swpackage -x media_type=tape -s /cdrom/QPK1100 QPK1100 @ \  
/var/tmp/QPK1100.depot
```

Entering this command copies the QPK1100 bundle contained in the depot to a file that can be uploaded into Opware SAS.

Example: File - Script to Split a Depot by Product

```
# This is an example script that splits a depot into individual
```

```
# product depots that can then be uploaded to the Opware
# Software Repository

for product in `swlist -l product -s <location of depot> | \
  cut -f1 | grep -v ^# | grep '[A-z]`
do
swpackage -x media_type=tape -s <location of depot> $product \
  @ /var/tmp/$product.depot
done
```

Example: File - Script to Split a Depot by Bundle

```
# This splits a depot into individual bundle depots that can
# then be uploaded to the Opware Software Repository

for bundle in `swlist -l bundle -s <location of depot> | \
  cut -f1 | grep -v ^# | grep '[A-z]`
do
swpackage -x media_type=tape -s <location of depot> $bundle \
  @ /var/tmp/$bundle.depot
done
```

Linux Packages

Linux packages are RPMs, which have the following characteristics:

- RPMs are both uploaded and installed as a unit so there is no distinction between container and installable packages.
- RPMs are versioned.

RPM Metadata

Opware SAS uses the metadata contained in RPMs when creating the package entries in the list of packages. An RPM contains the following metadata - the name, epoch, version, architecture and release of the RPM.

When a user uploads an RPM, Opware SAS performs the following actions:

- Opens the RPM and parses its metadata.
- Registers the RPM as an installable package.

When you upload a Linux RPM package to Opware SAS, the software policies related to that RPM may be updated. See “Install and Update of RPM Packages Using a Software Policy” on page 120 in this chapter for more information

Solaris Packages

Solaris packages are the container packages for Solaris. Solaris packages have the following characteristics:

- A Solaris package contains one or more package instances.
- When a Solaris package contains multiple instances, frequently only a subset of those instances will be installed because users might want to install only certain instances.
- Solaris packages have two basic formats:
 - **File system format:** The format for packages stored in a directory structure.
 - **Data stream format:** The format for standalone package files. This format is required for uploading Solaris packages into Opsware SAS.

The basic unit of Solaris packages is the package instance. Package instances have the following characteristics:

- Package instances are versioned.
- Users add package instances to a software policy. Opsware SAS adds package instances to and removes package instances from servers by using the `remediate` function. See the *Opsware® SAS User's Guide: Server Automation* for more information about `remediate`.

In the SAS Client, you can upload, view, download, and delete Solaris packages, and you can view, deprecate, and attach instances to software policies.

Opsware SAS supports Solaris packages in the following ways:

- Users upload Solaris packages in the uncompressed data stream file format.
- Opsware SAS can install interactive and non-interactive Solaris package instances. Interactive Solaris package instances require response files.
- Opsware SAS displays the name and version number for Solaris packages in the following way:

```
SUNW125f-1.0,REV=2001.03.21.17.00
SUNW1394h-11.9.0,REV=2002.04.06.15.27
```

- The Solaris utilities (such as `pkgadd`) use an admin file. The admin file stores settings regarding how the utilities should work. Each Opsware Agent on managed servers includes its own admin file that it uses when installing Solaris package instances. The admin file that the Opsware Agent uses is only used by Opsware SAS and does *not* set defaults for other applications using `pkgadd`.

- In some instances, a Solaris package might only get partially installed. A partial installation generally occurs when a package contains an installation script (other than the checkinstall script - for example, a preinstall or postinstall script) and that script exits non-zero during package installation. A partially installed Solaris package can be removed as if it were installed as a full package by removing it, or by overwriting it with a new package.
- For more information on `pkginfo`, `pkgadd`, and `pkgrm`, see the man pages.

Response files are text files. The entries in a response file occur as name = value pairs; for example, `BASEDIR="/opt/SUNWexplorer"` is a valid entry.

Opware SAS supports response files in the following ways:

- Users create response files outside of Opware SAS by using the `pkgask` Solaris utility.
- By using the Solaris Instance Package Properties page in the SAS Client users upload and overwrite the response files that are associated with Solaris package instances.
- Each response file is accessible only in the context of the Solaris package instance to which it belongs.
- Each Solaris package instance can have zero or one response file. Response files are not shared by different Solaris package instances.
- Attaching an interactive package to a software policy includes the response file because Opware SAS stores the response file with the package. You do not need to attach the response file to the software policy.
- After a Solaris package instance has a response file, Opware SAS uses that response file whenever the Solaris package instance is installed.
- If a Solaris package instance requires a response file and that file is missing in the SAS Client, Opware SAS might report an error when any server is remediated with that Solaris package instance.

When a user uploads a Solaris package, Opware SAS performs the following actions:

- Opens the package and parses its metadata.
- Automatically creates entries in the list of packages for the package instances in the package and registers them as installable.
- Registers the Solaris package as uninstallable.

Solaris Package Metadata

Opware SAS uses the metadata contained in Solaris packages when creating the package entries in the list of packages. A Solaris package contains the following metadata - the name, version, and description of each package instance in the package.

Prerequisites to Solaris Package Management

The Solaris package must be in data stream format before you can upload it to the Opware Software Repository. If it is in file system format, you can convert it by using the `pkgtrans` command:

```
pkgtrans -s <location of package> <new package> all
```

Windows Packages

Opware SAS supports the following Windows packages:

- Microsoft Installer Packages
- Microsoft Hotfixes, Security Patches, and Service Packs

Microsoft Installer Packages

Microsoft Installer packages (MSI) have the following characteristics:

- They contain all the information that the Microsoft Installer requires to install an application or product.
- They contain information that the installer requires to run the setup user interface.

MSI packages contain:

- An installation database
- A summary information stream
- Data streams for various parts of the installation

Opware SAS supports .msi files as installable software.

MSI Package Metadata

Opware SAS catalogs each MSI package by its ProductName and ProductVersion. These properties are defined in the Properties table of the MSI installation database.

Prerequisites to MSI Package Management

Opware SAS supports the Microsoft Windows Installer versions 1.1 and 2.0. Version 1.1 is included with Windows 2000, and version 2.0 is included with Windows 2003.

Windows NT does not include a version of the Windows Installer, but the Microsoft Windows redistributable can be obtained for download at <http://www.microsoft.com> or by including the `--withmsi` option on the Opware Agent Installer command line.

See the *Opware® SAS User's Guide: Server Automation* for more information about the steps to install an Opware Agent on a server.

Microsoft Hotfixes, Security Patches, and Service Packs

These packages include:

- Hotfixes
- Service Packs
- Security Patches

Hotfixes are issue specific and should only be applied if you experience the exact issue addressed by the hotfix, and only if you are using the current operating system version that has had the latest service pack applied.

Service packs are groups of hotfixes. They are more thoroughly tested than individually-released hotfixes, and are available to all customers, not just those with the specific problem.

Security patches are similar to hotfixes, but are mandatory if you are experiencing the specific problem they are created to address, and they need to be deployed as soon as they are made available.

When you upload a Service Pack, Opware requires the user to provide the version of the service pack. When you upload Hotfixes and Security Patches, Opware requires the user to provide the operating system version and the patch type.

ZIP Packages

ZIP Package Support

The SAS Client adds support for ZIP packages on the following operating systems:

- Windows
- Unix

ZIP Packaging

Use ZIP packages primarily to deliver code that can be run on a server. You can also use them to deliver application files for installing applications.

When a user installs a ZIP package on a server, the files are automatically extracted and saved to a directory that the user selects; otherwise, a default directory is used. Opsware SAS keeps track of all ZIP packages that it has installed, which prevents you from installing a ZIP package with the same name twice.

A ZIP package has no limits or restrictions on the size, format, or number of files that it contains.

Opsware SAS supports ZIP encapsulation for application package files that were built using other standalone installation programs, for example, InstallShield.

Opsware SAS requires silent install operation for programs designed for interactive installation. When you package these program files to upload to Opsware SAS, use the silent install options to play back automatic responses to provide unattended installation.

ZIP Packages Creation

Opsware SAS supports the ZIP file format for application package files that are built using non-MSI standalone installation programs, for example, InstallShield. Programs such as InstallShield were originally designed to provide for interactive installation. However, using the silent install feature, InstallShield users can play back a recording of a previous application installation that creates an unattended installation file with a suffix ISS.

The interactive installation recording is saved in the form of a setup.iss file that contains the responses to the interactive dialog boxes and popup menus that typically display during an interactive installation. After the response file is recorded, you can pass the setup.iss file as an argument to setup.exe executed from the command line to perform an unattended installation.

Similarly (using InstallShield), uninstalling an application can be set up to run unattended using the UnInst.exe command invoked with the -a and -y options to instruct the installer to run uninstall in silent mode.

See the documentation provided with your specific installer software for more information on silent install features and options.

Package Installation and Remove Scripts Definition

Because you can invoke silent installation and uninstallation from the command line, you can create scripts that perform silent installation and uninstallation. You can then include the scripts as part of the package file properties that you specify when you upload a ZIP file to the Software Repository.



When you specify the installation directory, post-installation, and pre-uninstallation script names for a ZIP package, do not use quotation marks to enclose the entire directory path and the script names.

When you uninstall a ZIP package, the extracted ZIP files that were installed on the server are not removed from the server. To uninstall those files, you must run an uninstallation script by specifying that script in the Pre-Uninstall Script Filename text box in the Package properties page.

Info-Zip Compatible ZIP Packages

Opware SAS offers package management support for Info-Zip compatible.zip packages. The files that are archived within Info-Zip are installable files on Opware SAS. You can download the.zip package creation tool from www.info-zip.org.

Info-Zip Compatible Package Metadata

Opware SAS uses the ZIP package file name to uniquely identify a ZIP package.

Prerequisites of Info-Zip Compatible Package Management

Full support for managing ZIP packages on a server is included with the Windows Opware Agent.

Windows Performance for Uploading Packages

When you upload packages from a Windows computer, users can improve the performance of the computer used to upload by changing TCP stack registry settings that affect upload speeds. The recommended change to the Windows registry file increases the default tcp-send buffer size from 8 KB to 16 KB.



Consult your system administrator before you make this change.

Perform the following steps to change the tcp-send buffer setting:

- 1** Using regedit, navigate to the following registry key:

```
HKEY_LOCAL_MACHINE
  SYSTEM
    CurrentControlSet
      Services
```

```
Afd
    Parameters (Create this key if it does not already
    exist)
```

- 2** Set the following value for the key:

```
Name: DefaultSendWindow
Value Type: REG_DWORD
Value: 16384 (decimal)
```

After you set the value, reboot the machine for the changes to take effect.

Character Encoding for Package Metadata and Scripts

In Opware SAS, you can specify the character encoding for package metadata and scripts in the following ways:

- Specify the encoding for package metadata when uploading packages in the SAS Client or by using the Opware Command Line Interface (OCLI).

When the encoding is specified, the SAS Web Client correctly displays in non-ASCII any package metadata, description fields, and error and status message returned by the operating system of the managed servers.

- Specify the encoding for scripts when uploading them in the SAS Web Client (in the Run Distributed Script Wizard and Scripts channel).

Opware SAS converts the script contents from the UTF-8 encoding to the encoding that you select. Internally, Opware SAS stores the script in the UTF-8 encoding.

After a script runs, you can download a ZIP file that contains the results encoded in UTF-8 format. For example, on Unix you can use the `iconv` program to interpret the downloaded results of the script execution.

The SAS Client includes the following selections for character encodings:

- Arabic (ISO-8859-6)
- Baltic (Cp1257)
- Baltic (ISO-8859-13)
- Baltic (ISO-8859-4)
- Central European (Cp1250)
- Central European (ISO-8859-2)
- Chinese Hong Kong, Taiwan (Cp950)

- Chinese Simplified (EUC-CN)
- Chinese Simplified (GB18030)
- Chinese Simplified (GBK)
- Chinese Traditional (Big5)
- Chinese Traditional (Big5-HKSCS)
- Chinese Traditional (EUC-TW)
- Cyrillic (Cp1251)
- Cyrillic (ISO-8859-5)
- Cyrillic (KOI8-R)
- English (US-ASCII)
- Greek (Cp1253)
- Greek (ISO-8859-7)
- Hebrew (Cp1255)
- Hebrew Visual (ISO-8859-8)
- Japanese (EUC-JP)
- Japanese (ISO-2022-JP)
- Japanese (Shift_JIS)
- Korean (Cp949)
- Korean (EUC-KR)
- Korean (JOHAB)
- South European (ISO-8859-3)
- Thai (TIS-620)
- Turkish (Cp1254)
- Turkish (ISO-8859-9)
- Unicode (UTF-8)
- Vietnamese (Cp1258)
- Western (Cp1252)

- Western (ISO-8859-1)
- Western (ISO-8859-15)

Importing a Package

Packages are downloaded from the vendor's web site and then imported (uploaded) into Opware SAS. A package can be imported with the SAS Client or by using OCLI (Version 1.0). See the *Opware® SAS Content Utilities Guide* for information about how to import packages by using OCLI.

If a package that is being uploaded already exists in the Software Repository, Opware SAS overwrites the package. If you upload Solaris patch clusters that contain patches that already exist in the Software Repository, the patches are overwritten. However, Opware SAS preserves any reboot options or flags set for the patches in the SAS Web Client.



You must have a set of permissions to import packages. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to import a package:

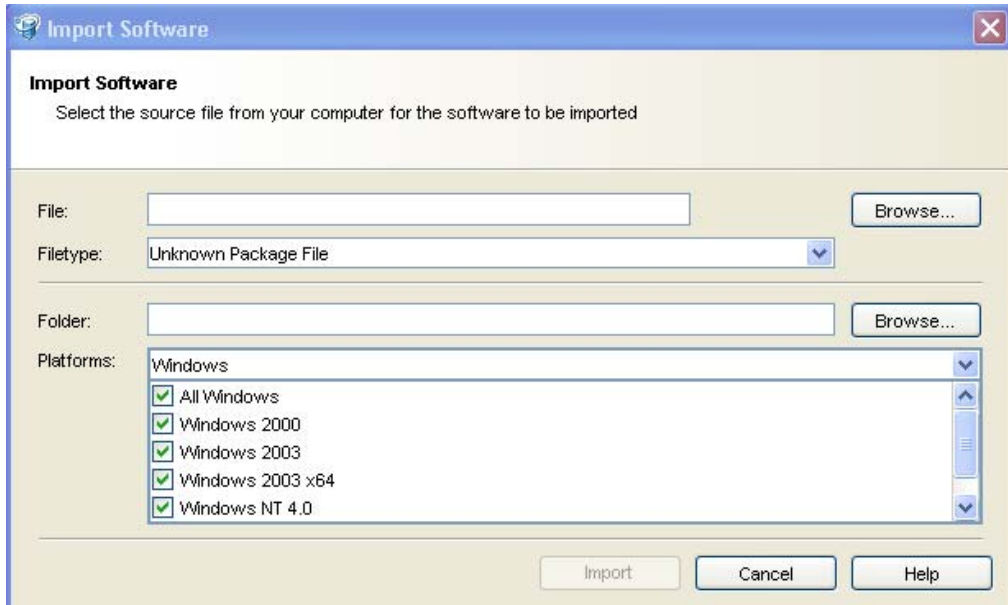
- 1** From the Navigation pane, select **Library ► By Type ► Packages**. The packages organized by operating system appear in the Content pane.

Or

From the Navigation pane, select **Library ► By Folder** and then select the folder in which the package should be located.

- From the **Actions** menu, select **Import Package**. The Import Software window appears as shown below.

Figure 2-13: Import Software Window in SAS Client



- Click **Browse** to locate and select the package to import.
- In the Open window, select the character encoding to be used by the package from the Encoding drop-down list.

You need to specify the character encoding so that Opware SAS can extract the metadata contained in the package and correctly display the information in non-ASCII characters in the SAS Client (for example, in the Package Properties pages). Package metadata includes comments, READMEs, scripts, descriptions, and content lists.

- In the Import Software window, select the file type from the Filetype drop-down list.
- Click **Browse** to specify the folder location for the package. The Select Location window appears. You cannot specify the folder location for Solaris Patches and Solaris Patch Clusters since they are not located in folders.
- From the Platform drop-down list, select the operating system.
- Click **Import**.

Exporting a Package

You can export (download) a package to your local computer so that you can check the installation of the package on a test or staging machine.



Package types that are not physical files like APARs cannot be downloaded.

Perform the following steps to download a package:

1 From the Navigation pane, select **Library** ► **By Type** ► **Packages**. The packages organized by operating systems appear in the Content pane.

Or

From the Navigation pane, select **Library** ► **By Folder** and then select the folder which contains the package.

2 From the Content pane, select a package to export.

3 From the **Actions** menu, select **Export Package**. The Export Software window appears.

4 In the Browse window, specify the location for the package to be exported to.

5 Click **Export**.

Ways to Open a Package


In the SAS Client, you can open a package in the following ways:

- Opening a package from Search
- Opening a package from the By Type view in the Library
- Opening a package from the Folder view in the Library

Opening a Package from Search

1 From the Navigation pane, select Search.

2 Select Software from the drop-down list and then enter the name of the package in the text field.

3 Select . The search results appear in the Content pane.

- 4 From the Content pane, select the package and then select **Open** from the **Actions** menu. The Package window appears.

Opening a Package from the By Type view in the Library

- 1 From the Navigation pane, select **Library > By Type > Packages**. The packages appear in the Content pane.
- 2 From the Content pane, select the package and then select **Open** from the **Actions** menu. The Package window appears.

Opening a Package from the By Folder view in the Library

- 1 From the Navigation pane, select **Library > By Folder**. The folder hierarchy in the Library appears in the Content pane.
- 2 From the Content pane, select the package in a folder and then select **Open** from the **Actions** menu. The Package window appears.

Viewing Package Properties

Perform the following steps to view the properties of a package:

- 1 From the Navigation pane, select **Library > By Type > Packages**. The packages organized by operating systems appear in the Content pane.

Or

From the Navigation pane, select **Library > By Folder** and then select the folder which contains the package.

- 2 From the Content pane, select the package to view.
- 3 From the **Actions** menu, select **Open**. The Package window appears.
- 4 From the Views pane, select Properties. The following package properties appear in the Content pane:

General Properties

- **Name:** The name of the package.
- **Description:** The description of the package's contents.
- **Type:** The type of package.
- **OS:** The operating systems associated with the package.
- **Location:** The location of the package in the folder hierarchy.
- **Install Path** (Only for Zip packages): The path where the package is installed on a

server.

- **Last Modified:** The date when the package was last modified.
- **Last Modified By:** The Opsware user who last modified the package.
- **Created:** The Opsware user who created the package.
- **Created By:** The date when the package was created.
- **File Name:** The file name of the package.
- **File version:** The file version of the package.
- **File Size:** The file size of the package.
- **Opsware ID:** The unique Opsware ID for the package.

Archived Scripts (Zip Packages only)

- **Post-Extraction Script:** The name of the post-extraction script to be run after installing the zip package.
- **Pre-Removal Script:** The name of the pre-removal script to be run before uninstalling the zip package.
- **If Script Returns Error:** An option that stops installation of the package if the script fails.

Install Parameters

- **Install Command:** The command that will be used to install the package.
- **Install Flags:** The optional arguments to be run when the package is installed on a managed server.
- **Reboot Required:** An option that reboots the server when the package is successfully installed.
- **Response File** (Only for Solaris packages): The Response files that are associated with Solaris package instances.
- **Upgrade** (Only for RPM packages): An option that runs the `-U` parameter during package installation.

Install Scripts

- **Pre-Install Script:** A script required to run on a managed server before the package is installed.
- **Post-Install Script:** A script required to run on a managed server after the package is installed.

- **Stop install if script returns an error:** An option that stops installation of the package if the script fails.

Uninstall Parameters

- **Uninstall Command:** The command that will be used to uninstall the package.
- **Uninstall Flags:** The optional arguments to be run when the package is uninstalled on servers.
- **Reboot Required:** An option that reboots the server when the package is successfully uninstalled.

Uninstall Scripts

- **Pre-Uninstall Script:** A script required to run on a managed server before the package is uninstalled.
- **Post-Uninstall Script:** A script required to run on a managed server after the package is uninstalled.
- **Stop uninstall if script returns an error:** An option that stops uninstallation of the package if the script fails.

Editing Package Properties

After you upload a new package or select an existing package, you can add or edit the package properties in the SAS Client.

You can edit a package's name, description, install parameters, install scripts, uninstall parameters, and uninstall scripts. You cannot change the operating system association of the package by editing the package properties.



You must have a set of permissions to edit the package properties. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

Perform the following steps to edit the properties of a package:

- 1** From the Navigation pane, select **Library ► By Type ► Packages**. The packages organized by operating system appears in the Content pane.

Or

From the Navigation pane, select **Library ► By Folder** and then select the folder that contains the package.

- 2** From the Content pane, select a package to edit.
- 3** From the **Actions** menu, select **Open**. The Package window appears.
- 4** From the Views pane, select Properties. The package properties will display in the Content pane.
- 5** Edit the following properties for the package:
 - **Name**: Specifies the name of the package.
 - **Description**: Specifies a short description that is used to indicate the package's contents.
 - **Availability**: Specifies the status of the package within Opsware SAS.
 - **Post-Extraction Script** (Only for ZIP packages): The name of the post-extraction script to be run after installing the zip package. Quotes in the file name must be preceded by backslashes.
 - **Pre-Removal Script** (Only for ZIP packages): The name of the pre-removal script to be run before uninstalling the zip package. Quotes in the file name must be preceded by backslashes.
 - **Install Flags**: Specifies the optional arguments to be run when the package is installed on servers.
 - **Reboot Required**: Selecting this option reboots the server when the package is successfully installed.
 - **Response File** (Only for Solaris packages): Specifies the Response files that are associated with the Solaris package instances.
 - **Upgrade** (Only for RPM packages): Selecting this option runs the `-U` parameter when the package is installed.
 - **Pre-Install Script**: Specifies the script required to run on a managed server before the package is installed.
 - **Post-Install Script**: Specifies the script required to run on a managed server after the package is installed.
 - **Stop install if script returns an error**: Selecting this option stops installation of the package if the script fails.
 - **Uninstall Flags**: Specifies the optional arguments to be run when the package is uninstalled on servers.

- **Reboot Required:** Selecting this option reboots the server when the package is successfully uninstalled.
- **Pre-Uninstall Script:** Specifies the script required to run on a managed server before the package is uninstalled.
- **Post-Uninstall Script:** Specifies the script required to run on a managed server after the package is uninstalled.
- **Stop uninstall if script returns an error:** Selecting this option stops uninstallation of the package if the script fails.

6 To save the changes, select **Save** from the **File** menu.

Viewing Package Contents

Perform the following steps to view the contents of a package:

1 From the Navigation pane, select **Library ► By Type ► Packages**. The packages organized by operating system appear in the Content pane.

Or

From the Navigation pane, select **Library ► By Folder** and select the folder which contains the package.

- 2** From the Content pane, select a package to view.
- 3** From the **Actions** menu, select **Open**. The Package window appears.
- 4** From the Views pane, select Contents. The package contents appears in the Content pane.
- 5** From the Content pane, select Files to display the list of files that will be installed by the package.
- 6** From the Content pane, select Scripts to display the list of scripts that will be executed by the package.



The package contents are only available for ZIP and RPM packages. For Solaris packages, HP-UX Depot, and AIX LPP, the package names of the children packages are displayed.

Viewing Servers Associated with a Package

Perform the following steps to view the servers on which the package is installed:

- 1 From the Navigation pane, select **Library** ► **By Type** ► **Packages**. The packages organized by operating system appear in the Content pane.
Or
From the Navigation pane, select **Library** ► **By Folder** and then select the folder which contains the package.
- 2 From the Content pane, select a package to view.
- 3 From the **Actions** menu, select **Open**. The Package window appears.
- 4 From the Views pane, select Server Usage. The list of servers associated with the package will display in Content pane.

Viewing All Software Policies Associated with a Package

Perform the following steps to view software policies which contain the package:

- 1 From the Navigation pane, select **Library** ► **By Type** ► **Packages**. The packages organized by operating system appear in the Content pane.
Or
From the Navigation pane select **Library** ► **By Folder** and select the folder which contains the package.
- 2 From the Content pane, select a package to view.
- 3 From the **Actions** menu, select **Open**. The Package window appears.
- 4 From the Views pane, select Software Policy Usage. The list of software policies associated with the package appears in Content pane.

Deleting a Package

Perform the following steps to delete a package:



You must have a set of permissions to delete a package. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

- 1 From the Navigation pane, select **Library ► By Type ► Packages**. The packages organized by operating system appear in the Content pane.

Or

From the Navigation pane, select **Library ► By Folder** and then select the folder which contains the package.

- 2 From the Content pane, select a package to delete.
- 3 From the **Actions** menu, select **Delete**.

Renaming a Package

Perform the following steps to rename a package:

- 1 From the Navigation pane, select **Library ► By Type ► Packages**. The packages organized by operating system appear in the Content pane.

Or

From the Navigation pane, select **Library ► By Folder** and select the folder which contains the package.

- 2 From the Content pane, select a package to rename.
- 3 From the **Actions** menu, select **Rename**. Enter the new name.
- 4 To save the changes, select **Save** from the **File** menu.

Locating Packages in Folders

Perform the following steps to locate a package in the folder hierarchy:

- 1 From the Navigation pane, select **Library ► By Type ► Packages**. The packages organized by operating system appear in the Content pane.

Or

From the Navigation pane, select **Library ► By Folder** and select the folder which contains the package.

- 2 From the Content pane, select the package and then select **Locate in Folders** from the **Actions** menu. The folder hierarchy for the package appears in the Content pane.

RPM Deployment

Opware SAS allows you to deploy RPM packages on Red Hat Linux and SUSE Linux servers without manually specifying all the dependent packages required for installing the RPM packages. When you deploy a RPM package, Opware SAS determines the dependencies and installation order for the RPM package, and identifies if any conflicts exists between the dependencies. After you resolve the conflicts, Opware SAS installs the RPM packages on the managed server.

In the SAS Client, you can install and uninstall RPM packages on Linux servers using software policies and also update the RPM packages in a software policy to their latest version. Opware SAS also allows you to automatically download the Red Hat Linux Errata into Opware and convert them to software policies. See “Automatically Importing Red Hat Network Errata” on page 127 in this chapter for more information.

RPM Deployment Process

Deploying RPM packages on Linux servers involves the following steps:

- Uploading RPM packages. See “Importing a Package” on page 109 in this chapter for information about uploading packages.
- Creating a software policy. See “Creating a Software Policy” on page 62 in this chapter for more information.
- Adding RPM packages to a software policy and then setting the installation and upgrade options for the RPM packages in the software policy. See “Adding an RPM Package to a Software Policy” on page 71 in this chapter for more information. See “Setting Installation and Update Options for an RPM Package” on page 77 in this chapter for more information.
- Attaching the software policy to a managed server. See the *Opware® SAS User's Guide: Application Automation* for information about attaching a software policy to a server.
- Remediating the server against the software policy. See the *Opware® SAS User's Guide: Application Automation* for information about remediating servers.

RPM Dependencies

After adding RPM packages to a software policy when you remediate the software policy on a managed server, Opware SAS identifies all the dependencies for the RPM packages specified in the software policy and the install order requirements necessary for the RPM package to be installed on the server. The dependencies include all the

packages that need to be installed or upgraded before or during the installation of the RPM package. Opsware SAS also analyzes the server's package inventory and identifies any conflicts between what is already installed and what needs to be installed.

During remediation in the preview remediate step, you can view the list of packages dependent on the RPM package to be installed and any conflicts between the dependencies. You can then resolve the dependencies when more than one RPM satisfies a dependency. After you resolve the dependencies, Opsware SAS installs the RPM packages specified in the software policy. See the *Opsware® SAS User's Guide: Application Automation* for more information about remediating software policies.

While remediating a server with software policies containing multiple versions of an RPM package, Opsware SAS will only install the latest version of the RPM package and its dependent packages.

If a software policy contains RPM packages and ZIP packages, then during remediation, Opsware SAS will first install the RPM packages, then the dependent packages, and finally all of the ZIP packages.



During remediation Opsware SAS does not support dependency solving for RPM packages for SUSE Linux Enterprise Server 8, SUSE Linux Standard Server 8 and all non Linux operating systems that support RPM packages.

Install and Update of RPM Packages Using a Software Policy

Opsware SAS allows you to install and update RPM packages on a server using software policies. After you import RPM packages to Opsware, you can add the RPM packages to the software policy. See “Importing a Package” on page 109 in this chapter for more information. See “Adding an RPM Package to a Software Policy” on page 71 in this chapter for more information.

In a software policy, you can specify whether the RPM packages listed in the software policy should be installed on the server or if the RPM package in the software policy should be updated to the latest version. In a software policy, you can set the following options for an RPM Package:

- Install Mode
- Version
- Release

The Install Mode option determines whether the RPM package listed in the software policy will be installed on the managed server. The Version and Release options determine whether the RPM package listed in the software policy will be updated to the newer release or version. See “Automatically Updating RPM Packages in a Software Policy” on page 121 in this chapter for more information. In addition, the Upgrade option for the RPM package in the Package Properties page determines if the RPM package will be updated. See “Upgrade Option For an RPM Package” on page 122 in this chapter for more information.

- **Install Mode:** If you select the **Update only** check box in the Install Mode, Opware SAS updates the RPM version on the managed server to the version specified in the software policy. This will happen when you remediate the software policy on the managed server.

If the RPM package is not already installed on the managed server, Opware SAS does not install the RPM version specified in the software policy.

If the **Update Only** check box is not selected, Opware SAS will install the RPM packages specified in the software policy on the managed server when you remediate the software policy on the managed server.

See “Setting Installation and Update Options for an RPM Package” on page 77 in this chapter for information about how to set these options.

Automatically Updating RPM Packages in a Software Policy

Opware SAS allows you to automatically update the version and/or release of the RPM packages in a software policy. In a software policy containing RPM packages, the Version and Release options determine whether the RPM package listed in the software policy will be updated to the newer release or version. Also the newer release or version of the RPM package must be placed in the same folder as the RPM package specified in the software policy.

Thus an RPM package in a software policy is automatically updated if the Version or Release option is selected for an RPM package in the software policy and if the newer version or release of the RPM package is present in the same folder as the RPM package in the software policy. The RPM packages in the folder are updated when you import an RPM package into the folder or when an RPM package is moved or copied from one folder to the other.

- **Version:** This option allows you to automatically update the version or release of the RPM packages in a software policy. If you select the **Auto** check box, Opware SAS will

automatically update the RPM packages in the software policy to a newer version or release of the RPM. To upgrade the RPM packages on a managed server, you must remediate the server with the software policy.



Opsware SAS will update the RPM package in the software policy to a newer version or release only if the newer release or version is placed in the same folder, as the RPM package specified in the software policy.

- **Release:** This option allows you to automatically update the version of the RPM package in a software policy. If you select the **Auto** check box, Opsware SAS will automatically update the RPM package in the software policy to a newer release of the same version of the RPM. To upgrade the RPM packages on a managed server, you must remediate the server with the software policy.



Opsware SAS will update the RPM package in the software policy to a newer release only if the newer release of the same version is placed in the same folder, as the RPM package specified in the software policy.

See “Setting Installation and Update Options for an RPM Package” on page 77 in this chapter for information about how to set these options.

Upgrade Option For an RPM Package

Upgrading a RPM package also depends on the Upgrade option for the RPM package (and the dependencies) specified in the Package properties window. After you upload an RPM package to Opsware SAS, you can set the set the Upgrade option for the RPM package in the Package properties Window. See “Editing Package Properties” on page 114 in this chapter for more information.

In the Package Properties window if you set the Upgrade option to Yes, then during remediation, Opsware SAS first removes the previous version of the RPM package and its dependencies from the server and then installs the newer version of the RPM package and its dependencies on the server.

In the Package Properties window if you set the Upgrade option to No, then during remediation Opsware SAS installs the newer version of the RPM package and its dependencies on the server. The previous version of the RPM package and its dependencies are not removed from the server.

Updating an RPM package thus depends on the options you set for an RPM package on the Software Policy window and the Package Properties window. The following table lists the action taken on the RPM package depending on the options set for the RPM package in the Software Policy window and the Package Properties window

Table 2-3: Setting Options for an RPM Package

OPTION SELECTED FOR AN RPM PACKAGE IN THE SOFTWARE POLICY WINDOW	OPTION SELECTED FOR AN RPM PACKAGE IN THE PACKAGE PROPERTIES WINDOW	ACTION TAKEN
The Update Only check box is selected in the Install Mode option.	The Upgrade option is set to Yes.	If a previous version of the RPM package is installed on the server, then Opsware SAS will install the newer version of the RPM package as specified in the software policy on the server and uninstall the previous version from the managed server. If the RPM package is not already installed on the server, then Opsware SAS will not install the RPM version specified in the software policy.

Table 2-3: Setting Options for an RPM Package (continued)

OPTION SELECTED FOR AN RPM PACKAGE IN THE SOFTWARE POLICY WINDOW	OPTION SELECTED FOR AN RPM PACKAGE IN THE PACKAGE PROPERTIES WINDOW	ACTION TAKEN
The Update Only check box is not selected in the Install Mode option.	The Upgrade option is set to Yes.	If the RPM package is not already installed on the managed server, then Opware SAS will install the RPM packages specified in the software policy on the managed server. If a previous version of the RPM package is installed on the server, then Opware SAS will install the newer version of the RPM package as specified in the software policy on the server and uninstall the previous version from the managed server.
The Update Only check box is selected in the Install Mode option.	The Upgrade option is set to No.	If a previous version of the RPM package is installed on the server, then Opware SAS will not install the newer version of the RPM package as specified in the software policy on the server. If the RPM package is not already installed on the server, then Opware SAS will not install the RPM version specified in the software policy.

Table 2-3: Setting Options for an RPM Package (continued)

OPTION SELECTED FOR AN RPM PACKAGE IN THE SOFTWARE POLICY WINDOW	OPTION SELECTED FOR AN RPM PACKAGE IN THE PACKAGE PROPERTIES WINDOW	ACTION TAKEN
The Update Only check box is not selected in the Install Mode option.	The Upgrade option is set to no.	If the RPM package is not already installed on the managed server, then Opsware SAS will install the RPM packages specified in the software policy on the managed server. If a previous version of the RPM package is installed on the server, then Opsware SAS will not install the newer version of the RPM package as specified in the software policy on the server.

In Opsware SAS, when you upload a non- kernel RPM package, by default the Upgrade option is set to Yes and when you upload a kernel RPM package, by default the Upgrade option is set to No. Therefore, when you remediate a server with a software policy containing kernel RPMS (such as kernel, kernel-bigmem, kernel-enterprise, kernel-smp, kernel-modules, kernel-debug, kernel-unsupported, kernel-source, kernel-devel), then Opsware SAS will always install the newer version of the kernel RPM packages and its dependencies on the server. The previous version of the kernel RPMS and its dependencies are not removed from the server.

Uninstalling RPM Packages

Opsware SAS allows you to uninstall RPM packages and downgrade to a previous version of the RPM package using software policies. To uninstall an RPM package from a managed server, you must first detach the software policy from the server and then remediate the server against the software policy. See the *Opsware® SAS User's Guide: Application Automation* for information on how to detach a software policy from a server.

When you remediate the server, Opware SAS uninstalls the RPM package specified in the software policy from the server and the dependent packages for the specified RPM package. You can uninstall the RPM packages only if they are not used by another software policy. When you uninstall an RPM package, Opware SAS also uninstalls any RPM packages which depend on the RPM packages being uninstalled.

Opware SAS also allows you to downgrade to a previous version of an RPM package using a software policy. To downgrade to a previous version of an RPM packages perform the following steps:

- 1** Detach the software policy containing the newer version of the RPM package from the server. See the *Opware® SAS User's Guide: Application Automation* for more information about detaching a software policy.
- 2** Remediate the server to uninstall the RPM package. See the *Opware® SAS User's Guide: Application Automation* for more information about remediating a server.
- 3** Create a new software policy. See “Creating a Software Policy” on page 62 in this chapter for information about creating a software policy.
- 4** Add the older version of the RPM package to the software policy. See “Adding an RPM Package to a Software Policy” on page 71 in this chapter for information about adding RPM packages.
- 5** Attach the software policy to the server. See the *Opware® SAS User's Guide: Application Automation* for more information about attaching a software policy.
- 6** Remediate the server to install the RPM package. See the *Opware® SAS User's Guide: Application Automation* for more information about remediating a server.

Software Policy Compliance for RPM Packages

A server can be either compliant or non-compliant with respect to a software policy attached to it. If the server's configuration does not match the packages, RPM packages, patches, and application configurations defined in a software policy (attached to that server), then the server is said to be non-compliant with that software policy. For RPM packages, software compliance is calculated based only on the RPM packages specified in the software policy. The dependent packages for RPM specified in the software policy are not used for calculating the software compliance.

See *Opware® SAS User's Guide: Application Automation* for more information about software policy compliance and how to perform a software policy scan.

Automatically Importing Red Hat Network Errata

Red Hat Network allows system administrators to manage their Red Hat servers on the network. Red Hat Linux publishes Errata which contains information describing security patches, bug fixes, and package updates for Red Hat Enterprise Linux. To install the packages in the Errata, the Errata must be downloaded from the Red Hat web site and imported into Opsware SAS. Using Opsware SAS you can automatically download the Errata released by Red Hat, convert them to software policies, and store the software policy in a folder in the Library in the SAS Client.

Red Hat Network also consists of channels which contain packages. Using Opsware SAS you can automatically download the packages in a channel, convert them to software policies, and store the software policy in a folder in the Library in the SAS Client

The `rhn_import` CLI program provided by Opsware enables you to create software policies, which correspond to Red Hat Network errata and channels. Using the `rhn_import` program, you can create the following types of software policies:

- **Channel based software policy:** A Red Hat Network channel contains a list of packages. A channel allows you group packages as per your organizational requirements. For example, a channel may contain packages for a particular Red Hat operating system version or architecture. A channel may contain other child channels. When you run the `rhn_import` program, Opsware SAS downloads the latest packages from the Red Hat Network channel and then imports the packages to the Library in the SAS Client and creates a channel based software policy. Thus, a channel based software policy reflects a particular channel. In the SAS Client, you can view the name, description, location, availability, and the operating system version of the channel based software policy in the Library. See “Viewing Errata Based and Channel Based Software Policies in the SAS Client” on page 128 in this chapter for more information.
- **Errata based software policy:** Red Hat Network Errata contains information on a particular problem and the associated packages to resolve the problem. An Errata based software policy contains all the individual erratum based software policies for a given channel. When you run the `rhn_import` program, Opsware SAS downloads the latest packages from the Red Hat Network errata and then imports the packages to the Library in the SAS Client and creates an errata based software policy. There are three types of Red Hat Network Errata: Bug Fix Advisories, Product Enhancement Advisories, and Security Advisories. The `rhn_import` program allows you to create errata software policies for Bug Fix Advisories, Product Enhancement Advisories, and Security Advisories in the SAS Client. In the SAS Client, you can view the name, description, location, availability, and the operating system version of the errata based

software policy in the Library. See “Viewing Errata Based and Channel Based Software Policies in the SAS Client” on page 128 in this chapter for more information.

- **Erratum based software policy:** Erratum based software policies contain packages associated with a particular erratum. When you run the `rhn_import` program, Opware SAS downloads the latest packages from the Red Hat Network erratum and then imports the packages to the Library in the SAS Client and creates an erratum based software policy.

To create and maintain software policies from the Red Hat Linux errata, erratum, and channels, log into the core server running the Software Repository component and run the `rhn_import` program located in the following directory:

```
/opt/opware/rhn_import/bin/rhn_import
```



Importing RPM packages from the Red Hat Network to Opware SAS requires a large amount of disk space. Over a period of time, the amount of disk space required increases as new versions of packages are released by the Red Hat Network. Opware recommends at least 5 GB of disk space is available in the Software Repository for every Red Hat Network channel you enable using the `rhn_import` program.

The documentation for the `rhn_import` program is available online. To view the complete documentation run the program with the following option:

```
/opt/opware/rhn_import/bin/rhn_import --manual
```

When you run the `rhn_import` program, you can specify the options listed the documentation provided online or use the Configuration File provided by Opware. The Configuration file provided by Opware with the `rhn_import` program is located in the following directory:

```
/etc/opt/opware/rhn_import/rhn_import.conf
```

Viewing Errata Based and Channel Based Software Policies in the SAS Client

The `rhn_import` program, allows you to create errata based, erratum based and channel based software policies in the SAS Client. After successfully running the program, you can view the properties of errata based, erratum based, and channel based software policies in the SAS Client. You can view properties such as the Opware user who created the software policy, the date when it was created, the name, the description, the

availability, the location of the software policy in the Library, the operating systems applicable to the software policy and the Opsware ID of the software policy. Opsware recommends that you do not edit the software policies which have been created by the `rh_import` program.



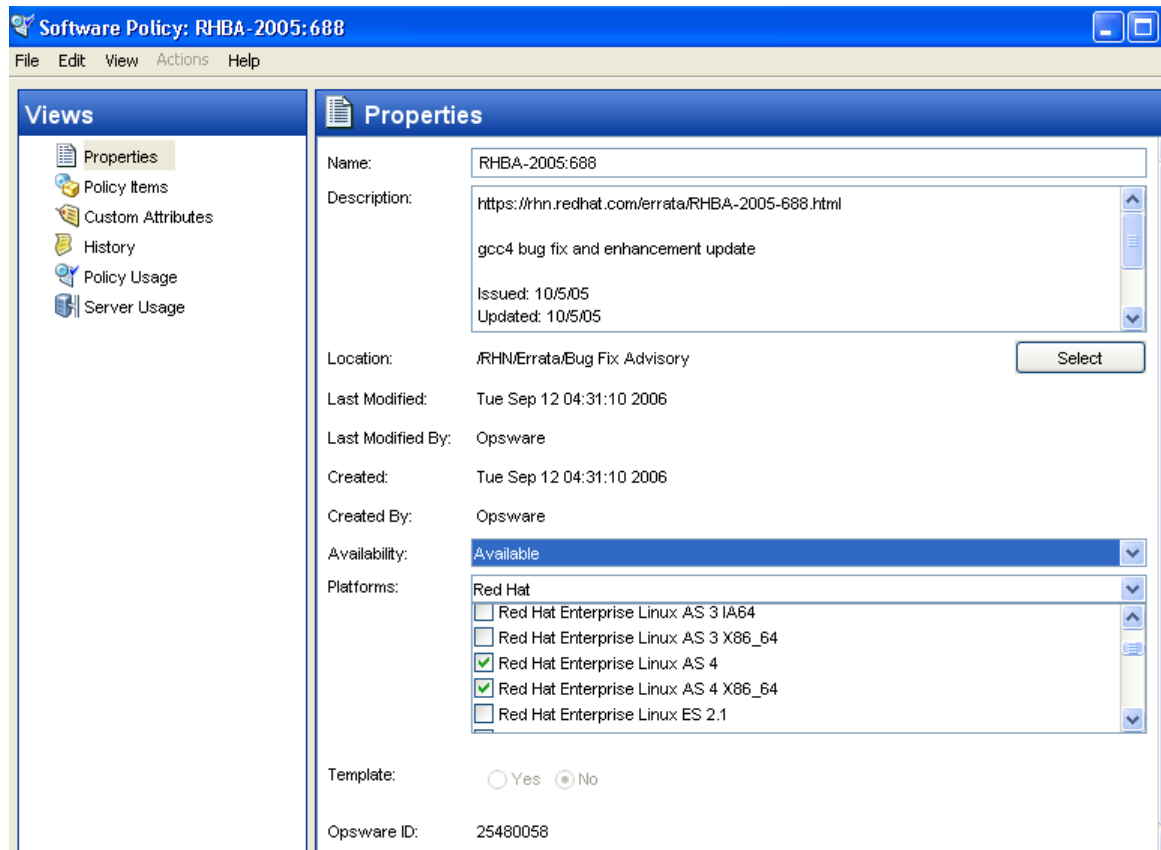
You must have a set of permissions to manage errata based, erratum based, and channel based software policies. To obtain these permissions, contact your Opsware administrator. See the *Opsware[®] SAS Administration Guide* for more information.

Perform the following steps to view the properties of a software policy:

- 1** From the Navigation pane, select **Library ► By Folder**.
- 2** Select the Red Hat Network Folder (RHN).

- 3 From the Content pane, select the errata based or channel based software policy and open it. The Software Policy window appears as shown in Figure 2-14.

Figure 2-14: Errata Software Policy Window



- 4 From the Views pane, select Properties. You can view the properties for the software policy in the Content pane.
 - **Name:** Contains the errata reference for the errata based software policy.
 - **Description:** Includes all the errata documentation for the errata.
 - **Location:** Specifies the location of the software policy in the folder hierarchy. To change the location click Select to specify the location for the software policy in the folder hierarchy. The Select Location window appears. Select a folder in the Library to specify the location of the software policy and then click **Select**.
 - **Created:** Corresponds to the time when the errata was downloaded by Opware to create the software policy.

- **Last Modified:** Corresponds to the time when the errata based software policy was modified.
 - **Availability:** Contains the Opsware server life cycle values for the errata based software policy. The default value for an errata based software policy is set to Available.
 - **Platform:** Specifies the all operating systems applicable to the errata.
- 5** To save the changes, select **Save** from the **File** menu.

Chapter 3: Operating System Provisioning Setup

IN THIS CHAPTER

This section discusses the following topics:

- OS Provisioning Setup
- Hardware Support in OS Provisioning
- OS Media Management
- OS Installation Profiles
- Using OS Installation Profiles
- Build Customization Scripts
- Default Custom Attribute Values
- Virtualization Support – VMware ESX (and Solaris 10)

This chapter contains information about setting up Opware SAS OS Provisioning, a feature that allows you to use the Opware SAS Client to install operating systems on bare metal and virtual servers as well as reprovision existing servers from boot images stored on a centralized media server.

Before you attempt to set up OS Provisioning, ensure that the OS provisioning components have been installed, configured correctly, and are available to you. Contact your Opware administrator for information about the installation and configuration of Opware SAS OS provisioning components.

OS Provisioning Setup

This section discusses the following topics:

- Overview
- OS Provisioning Setup Tasks
- Setting Up Sun Solaris OS Provisioning
- Setting Up Linux or VMware ESX OS Provisioning
- Setting Up Microsoft Windows OS Provisioning

Overview

Before you can begin provisioning servers with a new operating system, you must first set up OS Provisioning for each operating system you plan to install

This section explain how to set up the OS Provisioning infrastructure. You will also need to prepare an operating system installation profile for each OS. See “Overview” on page 149 for more information.

OS Provisioning supports installation-based provisioning using:

- Red Hat Linux Kickstart (for both Red Hat and VMware ESX)
- SUSE Linux YaST2
- Sun Solaris JumpStart
- Microsoft Windows unattended installation (image-based provisioning using WIM images is supported)

To prepare for OS provisioning, you, or your company's OS standards setter, should determine and record the standard configuration of each OS to be provisioned as well as the required utilities, drivers, and agents. System administrators can then use OS Provisioning to install the OSs, configure networking, and install other software.

OS Provisioning Setup Tasks



Before beginning, you must have the correct permissions to be able to set up OS Provisioning. You'll also need permissions to access the OS installation profiles. To obtain these permissions, contact your Opsware administrator. For more information about which permissions are required, see the *Opsware[®] SAS Administration Guide*.



You must have a licensed copy of the operating system installation media, which typically comes as a CD-ROM or DVD.

Although the details of OS Provisioning setup differs slightly depending on the OS you preparing for provisioning, the following tasks are representative of the setup process.

- Copy the OS media to the Media Server either from the licensed CD/DVD provided by the OS manufacturer or from any legal copy on your network.
- Create a Media Resource Locator (MRL) that points to the OS media on the Media Server by using the Opsware Import Media tool.
- Create a configuration file with a text editor that specifies how the OS will be installed.
- Prepare an OS Installation Profile in the SAS Web Client. Point to the location of the OS media by specifying in the profile the correct MRL.
- Upload the configuration file for use by OS Provisioning. See “Defining an OS Installation Profile – Unix” on page 155 or “Defining an OS Installation Profile – Windows” on page 159 for more information.
- Create an OS sequence in the SAS client.

Setting Up Sun Solaris OS Provisioning

OS Provisioning for Solaris includes a DHCP-based JumpStart configuration that hides the complexity of JumpStart from the end user. Unlike typical JumpStart systems, OS Provisioning does not require configuration updates to the JumpStart server for each installation that you provision.

Instead, you prepare an OS installation profile for each version of the Solaris OS to be installed in your environment.

The setup process for Solaris OS provisioning follows the general process for OS provisioning setup. However, you must perform certain setup tasks specifically for each Solaris OS version.

To set up Sun Solaris OS provisioning, perform the following tasks:

- 1** Copy the Sun Solaris OS media to the Media Server by using the scripts included on the Sun Solaris installation CD-ROM or DVD. See “Prerequisites for Creating an MRL” on page 141.
- 2** Create an MRL for the Solaris media by using the Import Media tool. See “Creating an MRL with the Import Media Tool” on page 146.
- 3** Create a Solaris profile with a text editor. See “Sun Solaris Profiles” on page 151.
- 4** Prepare an OS installation profile for the Solaris OS in the SAS Web Client. Specify the location of the Solaris OS media (with the MRL) and upload the profile. For more information, see “Defining an OS Installation Profile – Unix” on page 155.
- 5** (Optional) Customize the default build process that OS Provisioning uses to install the version of Solaris on servers. See “Solaris Build Customization Script” and “Requirements for Solaris Build Customization Scripts” on page 176.
- 6** (Optional) Edit the OS installation profile you have created for each version of Solaris, and see the following sections:
 - “Default Custom Attribute Values” on page 190.
 - “Custom Attributes for Sun Solaris” on page 191.
- 7** Create an OS Sequence in the SAS Client.

These sections explain how to configure aspects of the installation process so that it passes specific information to the Solaris build script.

Setting Up Linux or VMware ESX OS Provisioning

OS Provisioning for Linux includes a Kickstart and YaST2 system that hides the complexity of Kickstart and YaST2 from the end user.

VMware ESX provisioning is based on Red Hat's kickstart installation method and uses a kickstart configuration file, which specifies the choices you want to make during the installation of the VMware ESX Server software.

Unlike typical Kickstart or YaST2 systems, mapping a specific installation client to a particular configuration is a simple procedure. Opsware OS Provisioning allows each Linux OS (and template) to have a single configuration associated with it.

The setup for Linux OS provisioning follows the same general process for OS provisioning setup. However, you must perform certain setup tasks specifically for the Linux OS.

To set up Linux or VMWare ESX OS provisioning, perform the following tasks:

- 1** Copy the Linux or VMware ESX OS media to the Media Server. See “Prerequisites for Creating an MRL” on page 141 and “Creating an MRL with the Import Media Tool” on page 146.
- 2** Create a configuration file with a text editor. See the following sections:
 - “Red Hat Linux Configuration Files” on page 151
 - “VMware ESX Configuration Files” on page 152
 - “SUSE Linux Configuration Files” on page 152
- 3** Prepare an OS installation profile for the Linux or VMware ESX OS in the SAS Web Client. Specify the location of the Linux or VMware ESX OS media (with the MRL) and upload the configuration file. See “Defining an OS Installation Profile – Unix” on page 155.
- 4** (Optional) Customize the default build process that OS Provisioning uses to install Linux on servers. See the following sections for more information:
 - “Linux Build Customization Scripts” on page 181
 - “VMware ESX Build Customization Scripts” on page 182
 - “Requirements for Linux Build Customization Scripts” on page 181.
- 5** (Optional) Edit the OS installation profile you created for Linux or VMware ESX. See the following sections for more information:
 - “Default Custom Attribute Values” on page 190
 - “Custom Attributes for Linux or VMware ESX” on page 192

These sections explain how to configure aspects of the installation process so that it passes specific information to the Linux build script.

- 6 (Optional) Add new hardware support to a Linux or VMware ESX build image. OS Provisioning includes build images that install the target OS on servers for Linux or VMware ESX. For more information, see “Adding Hardware Support to a Linux or VMware ESX Build Image” on page 205.

Setting Up Microsoft Windows OS Provisioning

To prepare a Windows OS installation profile, you must set up a Windows unattended installation.

To set up Windows provisioning, you must have the following:

- A licensed copy of the Windows OS installation media, which typically comes as a CD-ROM or DVD.
- Mass storage drivers and Network Interface Card (NIC) drivers. The latest drivers can usually be downloaded from the hardware vendor's web site.
- A Windows setup response file.

The setup process for Windows OS provisioning follows the general process for OS provisioning setup. To set up Windows provisioning, perform the following tasks:

- 1 Copy the Windows OS media to the Media Server. See “Prerequisites for Creating an MRL” on page 141.
- 2 Create an MRL for the Windows media by using the Import Media tool. See “Creating an MRL with the Import Media Tool” on page 146.
- 3 Create a Windows response file with a text editor. See “Microsoft Windows Response Files” on page 153.
- 4 Prepare an OS installation profile for the Windows OS in the SAS Web Client. See “” on page 159. This section explains how to specify the location of the Windows OS media (with the MRL) and upload the response file.
- 5 (Optional) In the OS installation profile, upload hardware-specific files for the hardware you expect to provision by mapping a signature for that hardware to the correct hardware-specific profile.

OS Provisioning will select the correct Hardware Signature file at build time based on the hardware signature of the server that is about to be provisioned. For more information, see “Hardware Signature Files for Windows” on page 158.

- 6 (Optional) Customize the default build process that OS Provisioning uses to install the version of Windows on servers.

- 7** Edit the OS installation profile you created for each version of Windows.

In the installation profile, you can edit the custom attributes so that the profile passes specific information to the Windows build script to configure aspects of the installation process. You can also set a value for the timeout custom attribute. Setting this value controls the timeout value after an error.

For more information, see the following sections:

- “Default Custom Attribute Values” on page 190.
- “Custom Attributes for Microsoft Windows” on page 193.

- 8** (Optional) Create a Windows boot floppy. See “Creating a Windows Boot Image” on page 203. You will typically use this feature when you need to boot x86-process based servers from a floppy (i.e. servers that cannot be booted over a network).

- 9** (Optional) Add new hardware support to the Windows boot images. The default boot images for Windows include common NIC drivers for many hardware makes and models. Opware SAS uses these NIC drivers to boot new x86-processor-based servers for the first time.

For more information, see “Adding NIC Support to a Windows Boot Image” on page 201.

OS Media Management

This section discusses the following topics:

- Overview
- Prerequisites for Creating an MRL
- Setting Up the Media Server
- Import Media Tool Syntax and Options
- Creating an MRL with the Import Media Tool
- Editing an MRL
- Deleting an MRL

Overview

In order for OS Provisioning to be able to access the OS media, you need to copy it to the Opware Media Server. The Media Server provides access to the OS media over the network by using NFS for Linux, VMware ESX, and Solaris OS and SMB for Windows. After copying the OS media to the Media Server, you must “import” it into Opware SAS by running the Opware Import Media tool (a utility script included with Opware SAS). A single copy of the OS media on the Media Server can be used to provision multiple servers as long as you have valid licenses or license keys.

In this guide, the term OS media means the installation software for an OS from the software vendor. Typically, OS media is distributed on CD-ROM, DVD, or by downloading the software distribution from the vendor's FTP site. OS media can contain binaries for installing the OS, packages of different types, metadata about the packages, and other information.

The Role of the Import Media Tool

“Importing OS media” means that the Import Media tool creates an Opware-generated string called a Media Resource Locator (MRL) for each OS media that you want to provision that points to the OS media's location on the Media Server. The MRL is used by the Software Repository to identify the location of the OS media on the Media Server. Import media also uploads software packages related to the OS media to the Software Repository.

An MRL is a network path (in URI format) to the installation media for an OS on the Opware Media Server. When a server is being provisioned with an OS, the server mounts the network path for the OS media by using NFS (for Linux and Solaris), or SMB (for Windows). The MRL is registered with Opware SAS. An MRL should resolve to the Media Server in the local facility where Opware SAS is installed.

When you run the Import Media tool to create an MRL, the tool:

- Mounts the media at the specified network path by using NFS or SMB.
- Detects the OS (Solaris, Linux, VMware ESX, or Windows) and version of the media.
- Creates that MRL in Opware SAS based on the server name and path that you specify, so that you can use it in OS installation profiles.

- Uploads all packages to the Software Repository so that OS Provisioning can install them after initial OS provisioning. The default is `--upload = yes`. You can specify `-upload = no` if you do not want to upload all packages to the Software Repository.

The `--folder` option allows you to specify the full path to upload the OS media packages. This path corresponds to a folder inside the Library in the SAS Client. These packages can be added to a software policy in the SAS Client. The software policies can be associated with an OS sequence. After OS provisioning completes, the policies will be attached to the server and remediated. If you do not use the `--folder` option, then the packages will by default be uploaded to `/Package Repository/OS Media/<Platform Name>`.

Re-running the Import Media tool with the same server and path as an existing MRL updates the MRL, but does *not* re-upload duplicate Linux, Solaris, or VMware ESX packages.

For Linux, VMware ESX, and Microsoft Windows, the Import Media tool modifies portions of the OS media to integrate OS Provisioning with the vendor provisioning boot process.

Prerequisites for Creating an MRL

Before you run the Import Media tool, the OS media that you want to import must be available through the network on the Media Server. (If necessary, contact your Opsware administrator for the host name of the Media Server.)

You must know what locations were specified for the OS media. When Opsware SAS was installed, the Opsware Installer prompted for the path names of the root directories for the Windows, Solaris, Linux, and VMware ESX OS media on the Opsware Media Server. If necessary, contact your Opsware administrator for this information.

Before you perform the tasks to set up OS provisioning, you must have a licensed copy of the OS installation media, which typically comes as a CD-ROM or DVD.

Also, you will need to provide Opsware user credentials (username and password). This will enable you to use the import media tool to create an MRL and upload packages. Unless specified in the import media argument, you will be prompted for a valid user name and password when you execute the command.

Setting Up the Media Server

Perform the following tasks to set up the Media Server for OS provisioning:

- 1** On the Media Server host, create the directory structure for the versions of the OS that you plan to use for server provisioning.

Create the directory structure based on the root directories specified for the OS media during Opware SAS installation. If necessary, contact your Opware administrator for the locations of the OS media root directories.

- 2** Ensure that the media for each OS that you want to provision is available on the Media Server. Use the following guidelines:
 - *Microsoft Windows*: copy the OS media files to the location on the Media Server specified during the Opware SAS installation. If necessary, contact your Opware administrator for this information.
 - *Microsoft Windows Server 2003 Server x64*: copy the OS media files to the location on the Media Server specified during the Opware SAS installation. If necessary, contact your Opware administrator for this information. You then need to cd into that directory and issue these commands:

```
tar cf i386.tar i386
tar cf amd64.tar amd64
chmod a+r i386.tar amd64.tar [Makes files readable for all]
```
 - *Linux or VMware ESX*: copy the OS media files to the location on the Media Server specified during the Opware SAS installation. If necessary, contact your Opware administrator for this information. Note that, the OS media must be NFS exported read/write.
 - *SUSE Linux*: copy the OS media files to the location on the Media Server specified during the Opware SAS installation. If necessary, contact your Opware administrator for this information. If required, See <http://www.suse.de/~nashif/autoinstall/multiplesource.html> for information on how to deal with multiple sources.
 - *Sun Solaris*: use the Sun Solaris scripts included on the CD-ROM or DVD to copy the OS media files to the correct location on the Media Server.

Windows Media: Preparing Network Driver Directories

To ensure that the server you want to provision has the appropriate network card drivers for Windows 2000, 2003, and/or XP, you must create directories for those drivers on the Opware media server.

To create these directories on the Opware media server, perform the following tasks:

- 1** Log on to Opware media server as root.
- 2** Navigate to `windows_media_share/i386` and create the following directory:

```
$OEM$/$/Drivers/nic
```

- 3 Create a subdirectory to which downloaded driver files will be saved. Name the subdirectories in a way that will identify the drivers they contain. For example:
SC1425
- 4 Grant at least 755 permissions to the newly created directory and subdirectories.
- 5 Copy the driver files to the newly created directory.
- 6 If you need to specify OEM drivers, add a line similar to the following in the [Unattended] section of the unattend.txt file and reference the directory win which you are storing the drivers. For example:

```
OEMPnPDriversPath = "Drivers\NIC;Drivers\NIC\SC1425"
```

For more information about drivers, refer to <http://support.microsoft.com>.

Import Media Tool Syntax and Options

The following section provides the syntax and command line options for the Import Media tool.

To start the tool, log onto the Software Repository server and enter:

```
import_media [options] <network path>
```

The following networks paths are valid:

- NFS: `nfs://<NFS server>/<exported path>`
- Windows SMB: `smb://<SMB Server>/<share>/<path>/i386`

If the path contains spaces or shell metacharacters, it must be placed in quotes so that the shell passes it to `import_media` as a single argument.

To use a Windows SMB share, note the following:

- The Media Server directory where the OS media is mounted must comply with the conventions for a FAT file system. A directory name can consist of any combination (up to eight characters) of letters, digits, or the following special characters: `$ % Ã,Â´ – _ @ { } ~ ` ! # ()`. The directory name can also have an extension (up to three characters) of any combination of letters, digits, or the previously listed special characters. The extension is preceded by a period.

- A password was set for the root user (parameter: `media_server.windows_share_password`) when Opware SAS was installed to write-protect the Windows media share. The Opware Import Media tool prompts for the password each time you run it. Contact your Opware administrator for this password

Table 3-1 lists the command line options available for the import media command.

Figure 3-1: Import Media Tool Command Line Options

IMPORT MEDIA TOOL OPTION	DESCRIPTION
<code>--help</code>	Display this help.
<code>--folder</code>	Override Folder location. Default is: <code>"/Package Repository/OS Media/<Platform Name>".</code>
<code>--medianame=<displayname></code>	Override Opware-generated display name. Note: Use <code>'_'</code> to escape spaces in the name.
<code>--opsware-username</code>	Username for authenticating to Opware. If you do not supply <code>--opsware-username</code> on the command line, you will be prompted to enter it. If you do not have a valid Opware user name and password, contact your Opware administrator.
<code>--opsware-password</code>	Password for the Opware username. Warning: This option is not recommended, since passing passwords as command line options is insecure. When this option is omitted, the user will be prompted for the password securely.
<code>--mrl=<mrl></code>	Override automatic OS Media path generation. <code>--mrl=//MEDIA/PUB/WINNT/SERVER/I386</code> <code>--mrl=nfs://media/export/media/redhat/7.2</code>
<code>--smbuser=<user></code>	User for SMB access. Default is "root".

Figure 3-1: Import Media Tool Command Line Options (continued)

IMPORT MEDIA TOOL OPTION	DESCRIPTION
<code>--smbpasswd=<password></code>	Use this password for SMB access. Note: This will appear in cleartext on the command line. Warning: This option is not recommended, since passing passwords as command line options is insecure. When this option is omitted, the user will be prompted for the password securely.
<code>--logfile=<logfile></code>	Override log file location. Default is: <code>/var/log/opsware/mm_wordbot/import_media.log</code>
<code>--wimimage</code>	The path supplied refers to a (WIM) image. Be sure to also supply <code>--platform=<platform></code> , since the target platform cannot be autodetected.
<code>--platform=<platform></code>	Override automatic platform detection. Must match an existing Opsware platform defined in the Model Repository.
<code>--progress= [yes]</code>	Toggle display of progress (default is yes). For example: <code>--progress=no</code>
<code>--resolve-symlinks= [yes]</code>	Toggle resolution of symlinks (default is yes).
<code>--upload = [yes]</code>	Uploads all packages to the Software Repository so that OS Provisioning can install them after initial OS provisioning (default is yes).

Creating an MRL with the Import Media Tool

Perform the following steps to create an MRL with the Import Media tool:

- 1** Log into the Software Repository host as root.
- 2** Change to the following directory (where the Software Repository is located):
`/opt/opsware/mm_wordbot/util`
- 3** Ensure that you have the correct path to the directory into which you uploaded the OS media on the OS Media Server.
- 4** Run the following `import_media` script:

```
import_media [options] <network path>
```

For example, to import OS Windows media from an SMB share named OSMEDIA on the server `mediasrv`, enter:

```
import_media smb://mediasrv/OSMEDIA/WINNT/SERVER/I386
```

To import Linux (or VMware ESX) media from an NFS server named `mediaserver.company.com`, enter:

```
import_media nfs://mediaserver.company.com/export/media/redhat/7.2
```

To import Solaris media from an NFS server named `mediaserver.company.com`, enter:

```
import_media nfs://mediaserver.company.com/export/media/solaris/sol-10-u8-sparc
```

Unless otherwise specified, the default folder location for uploaded software packages is in the form `/Package Repository/OS Media/<Platform Name>`, where `<Platform Name>` is the (full) Opsware name for the platform detected in the media being imported. If the folder does not exist, then it will be created. To manually specify a folder location, use the `--folder` option.

Running the Import Media tool writes progress to the log file `import_media.log`. The log file is located on the server where you are running the Import Media tool script in the directory from which you invoke the script.

For information on the command line options for the Import Media tool, see “Import Media Tool Syntax and Options” on page 143.

Editing an MRL

Perform the following steps to edit an MRL:

- 1** Log into the SAS Web Client. The SAS Web Client home page appears.
- 2** From the **Navigation** pane, click Software ► Operating Systems. The **Operating Systems** page appears.
- 3** Select the **OS Media** tab. A list of Media Resource Locators (MRLs) appears.

Each MRL represents media available for installation. See Figure 3-1.

Figure 3-1: OS Media Page in the SAS Web Client

Delete...		1 Total		Windows NT 4.0	
<input type="checkbox"/>	Name	Path	OS Version	Size	Modified
<input type="checkbox"/>	Windows NT 4.0	theword\media\winnt4\i386	Windows NT 4.0	85.07 MB	04-22-2005

Path on the Media Server to the OS Media

- 4** Click the display name for the MRL that you want to edit. The **Edit OS Media** page appears, as Figure 3-2 shows.

Figure 3-2: Edit OS Media Page in the SAS Web Client

Name:	Red Hat Enterprise Linux AS 3
Description:	Red Hat Enterprise Linux AS 3 Media
OS Version:	Red Hat Enterprise Linux AS 3
Path:	nfs://mediaserver.c76.dev.opsware.com/media/op
Size:	1.53 GB
Last Modified:	Mon Feb 12 10:48:13 2007
ID:	38360076
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

- 5** You can modify the name, description, or path of the MRL.
- 6** Click **Save**.

Deleting an MRL

Note that, you cannot delete an MRL with the SAS Web Client when the MRL has been previously specified in an OS installation profile. To delete an MRL specified in an OS installation profile, you must first delete the OS installation profile or specify another MRL in the OS installation profile.

See “Defining an OS Installation Profile – Unix” on page 155 or “” on page 159 for more information.

Perform the following steps to delete an MRL:

- 1** Log into the SAS Web Client. The SAS Web Client home page appears.
- 2** From the **Navigation** pane, click Software ► Operating Systems. The Operating Systems page appears.
- 3** Select the **OS Media** tab. The list of media available for installation appears.
- 4** Select the OS Media that you want to delete.
- 5** Click **Delete**. (If the MRL is specified in an OS installation profile, a warning message appears.) The list of Media Resource Locators re-appears.

OS Installation Profiles

This section discusses the following topics:

- Overview
- Specifying Software in OS Installation Profiles
- Configuration Files
- Sun Solaris Profiles
- Red Hat Linux Configuration Files
- VMware ESX Configuration Files
- SUSE Linux Configuration Files
- Microsoft Windows Response Files

Overview

Before you create your Operating System Installation Profiles, you should have set up OS Provisioning and created MRLs pointing to the OS media using the Import Media tool. See “OS Provisioning Setup Tasks” on page 135 for the overall process of setting up OS provisioning.

OS Installation Profiles store all the relevant information required to provision an OS.

You create OS Installation Profiles by using the Prepare Operating System Wizard in the SAS Web Client.

The process of creating an Operating System Installation Profile includes:

- 1** Specifying properties for the OS.
- 2** Specifying the OS media from which to perform an installation by selecting an MRL. (See “OS Media Management” on page 139 for more information on editing MRLs.)
- 3** Uploading the following installation resources used during unattended installation:
 - A standard configuration file for the OS. (See “Configuration Files” on page 151 for more information.)
 - A build customization script, which can modify the installation process at certain points. (See “Build Customization Scripts” on page 171 for more information.)
 - *Microsoft Windows Only*: a Hardware Signature, which contains hardware specific information. (“Hardware Signature Files for Windows” on page 158 for more information.)

Table 3-4 compares the installation resources across operating systems.

Table 3-4: Installation Resources for OS Installation Profiles

INSTALLATION RESOURCE	SUSE	WINDOWS	SOLARIS	LINUX OR VMWARE ESX
Configuration File	Required autoinst.xml	Required unattend.txt	Required profile	Required profile

Table 3-4: Installation Resources for OS Installation Profiles (continued)

INSTALLATION RESOURCE	SUSE	WINDOWS	SOLARIS	LINUX OR VMWARE ESX
Build Customization Script	Optional executable file: bcs.tgz containing "run" script	Optional executable file: Windows DOS: bcs.cab containing "runphase.bat" script WinPE: bcs.zip containing "runphase.bat" script	Optional executable file: bcs.tar.Z containing "run" script	Optional Executable file: bcs.tgz containing "run" script
Hardware Signature File	Not required	Optional filename.txt	Not required	Not required



The configuration file that you upload for each OS can have any file name, however, when the file is uploaded, OS Provisioning renames the file so that it has the correct name for that OS.

You can edit an OS Installation Profile later to add support for new hardware or to change the way the OS is installed. See "Modifying Existing OS Installation Profiles" on page 163 in this chapter for more information.

Specifying Software in OS Installation Profiles

You can specify the packages to install during OS provisioning in the following ways:

- By uploading a configuration file that specifies to the vendor installation program the software packages to install.
- By creating software policies in the SAS Client, adding the desired packages, then associating the software policies with an OS Sequence.

Configuration Files

A configuration file is required for each OS Installation Profile:

- *Solaris*: you must create and upload a JumpStart profile.
- *Red Hat Linux or VMware ESX*: you must create and upload a Kickstart configuration file.
- *SUSE Linux*: you must create and upload a YaST2 configuration file.
- *Windows*: you must create and upload a response file.



If your configuration file will enable a firewall, you must ensure that all necessary ports and protocols for communication between the Opsware core and the Opsware OS Build Agent and the Opsware Agent are allowed. Refer to the *Opsware® SAS Planning and Installation Guide* for details. To help isolate firewall related issues, Opsware recommends leaving firewalls disabled while configuring OS provisioning for the first time and reenabling them once the system is correctly configured.

Sun Solaris Profiles

When preparing a Solaris OS installation profile, OS Provisioning requires that you upload a JumpStart profile.

The Solaris profile must:

- Be a valid profile that you would use with a JumpStart server.
- Specify that the installation type is an initial installation and not an upgrade.
- Specify a package-based installation by listing the clusters and packages to install.
- Specify disk partitioning information.

Red Hat Linux Configuration Files

The Red Hat Linux configuration file instructs the Kickstart server on the packages to install, how to partition the drive, and how to configure the runtime network post-installation.

When preparing a Red Hat Linux OS installation profile, Opsware SAS validates the Kickstart configuration file. When the configuration file is uploaded, OS Provisioning parses the file in order to extract the package list.

The Red Hat Linux configuration file must:

- Be a valid configuration file that you would use with a Kickstart server.
- Specify the RPM packages to install.
- Include the reboot option.

VMware ESX Configuration Files

VMware ESX provisioning uses a kickstart configuration file. This file consists of several VMware ESX Server installation parameters. You can configure this file to instruct the Kickstart server to install packages, to partition the drive, to configure the runtime network post-installation, etc.

The VMware ESX Kickstart configuration file must:

- Be a valid configuration file that you would use with a Kickstart server.
- Specify the RPM packages to install.
- Include the reboot option.

The VMware ESX Server provides a Web-based wizard (VI Web Access). Its web wizard interviews you for configuration information and then generates a configuration file.

For VMware ESX-specific commands that must appear in the configuration file and information about the configuration file wizard, see the VMware *Installation and Upgrade Guide*: "Remote and Scripted Installations". You can find this guide at <http://www.vmware.com>.

SUSE Linux Configuration Files

The SUSE Linux configuration file specifies to YaST2 which packages to install, how to partition the drive, and the OS configuration.

When preparing a SUSE Linux OS installation profile, Opware SAS validates the YaST2 configuration file. When the configuration file is uploaded, OS Provisioning parses the file and extracts the package list.

The SUSE Linux configuration file must:

- Be a valid YaST2 configuration file.
- Include the Reboot option and have the Confirm Properties option in the mode resource set to FALSE.

For SUSE Linux, see <http://www.suse.com/~ug/> for more information on installation.

Microsoft Windows Response Files

If you are creating a Windows OS installation profile, the configuration file must be an unattended installation response file that conforms to the following:

- The `OemPreInstall` key must be set to YES. If this key is not set, OS Provisioning will set it automatically.
- A network configuration must be specified so that when the OS boots for the first time, it will get a valid IP address.
- Any dialog boxes that might appear during the Text and GUI mode portions of Windows setup must be set so that they do not appear during the OS provisioning process.

When uploading an `unattend.txt` file, Opsware SAS validates the response file and rejects incomplete response files.

See “Sample Response File for Windows 2000” below and “Sample Response File for Windows Server 2003” on page 154 for examples of valid Windows response files.

Sample Response File for Windows 2000

The following sample response file shows typical valid responses for a Windows 2000 installation. This sample response file contains the required settings for Windows 2000 provisioning with OS Provisioning.

```
[Data]
  AutoPartition=0
  MsDosInitiated=0
  UnattendedInstall=Yes

[GuiUnattended]
  AdminPassword=3mbree0
  OEMSkipRegional=1
  OEMSkipWelcome=1
  ;004 Pacific Standard Time (GMT-08:00) Pacific Time (US and
  Canada); Tijuana
  ;See http://unattended.sourceforge.net/timezones.php
  TimeZone=004

[Identification]
  JoinWorkgroup=WORKGROUP

[LicenseFilePrintData]
  AutoMode=PerServer
```

```
AutoUsers=9999

[Networking]

[Unattended]
ExtendOemPartition=1
FileSystem=ConvertNTFS
OemPnPDriversPath=drivers\nic\intel
OemPreinstall=Yes
OemSkipEula=Yes
TargetPath=*
UnattendMode=FullUnattended

[UserData]
ComputerName=*
FullName="Windows 2000 Advanced Server"
OrgName=Opware
ProductID=XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Sample Response File for Windows Server 2003

The following sample response file shows typical valid responses for a Windows Server 2003 installation. This sample response file contains the required settings for Windows Server 2003 provisioning with OS Provisioning.

```
[Data]
AutoPartition=0
MsDosInitiated=0
UnattendedInstall=Yes

[GuiUnattended]
AdminPassword=3mbree0
OEMSkipRegional=1
OEMSkipWelcome=1
;004 Pacific Standard Time (GMT-08:00) Pacific Time (US and
Canada); Tijuana
;See http://unattended.sourceforge.net/timezones.php
TimeZone=004

[Identification]
JoinWorkgroup=WORKGROUP

[LicenseFilePrintData]
AutoMode = PerSeat
```

```
[Networking]
```

```
[Unattended]
```

```
ExtendOemPartition=1  
FileSystem=ConvertNTFS  
OemPnPDriversPath=drivers\nic\intel  
OemPreinstall=Yes  
OemSkipEula=Yes  
UnattendMode=FullUnattended
```

```
[UserData]
```

```
ComputerName=*  
FullName="Windows Server 2003"  
ProductKey=XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX
```

Using OS Installation Profiles

This section discusses the following topics:

- Defining an OS Installation Profile – Unix
- Hardware Signature Files for Windows
- Modifying Existing OS Installation Profiles
- Changing the OS Installation Profile Properties
- Modifying How an OS Is Installed on a Server – Unix
- Modifying the OS Installation Profile Packages
- Viewing Change History for an OS Installation Profile
- Deleting an OS Installation Profile

Defining an OS Installation Profile – Unix

To use the Prepare Operating System Wizard to define a Unix OS Installation Profile, perform the following steps:

- 1** Access Prepare Operating System window from the Opsware SAS Web Client or from the Opsware SAS Client:
 - *SAS Web Client* Home Page: click the **Prepare OS link** in the **Tasks** panel. Or, from the **Navigation** pane, click Software ► Operating Systems. The **Operating Systems** page appears. Click **Prepare OS**.

- **SAS Client:** from the **Navigation** pane, select Library ► OS Installation Profiles. Select an OS, then from the **Actions** menu, select **Create New**.
(If asked to log into the SAS Web Client, enter your user name and password, and click **Log In**.) The Describe OS page appears, as Figure 3-3 shows.

Figure 3-3: Describe OS Page in the Prepare Operating System Wizard

Prepare Operating System

1 Describe OS

2 Define Installation

3 Upload File

4 Review Packages

Describe OS

Enter the following information to describe the operating system.

Name:

Description:

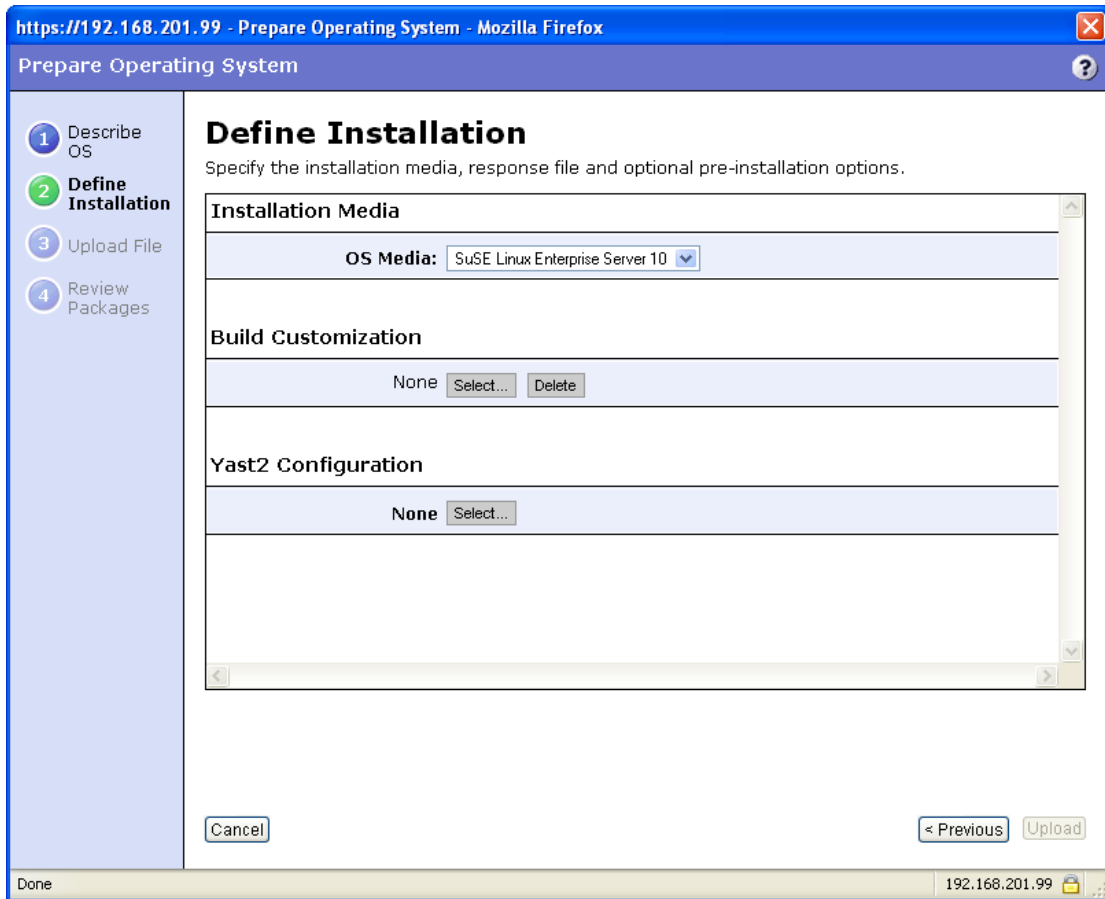
Customer:

OS Version:

- 2** Describe the OS by specifying the following information:
- **Name:** *(Required)* Specify the display name for the Unix OS.
 - **Customer:** *(Required)* Associate the Unix OS with a specific customer; to set up the OS for use by all customers, select “Customer Independent”.
 - **OS Version:** *(Required)* Specify the version of the Unix OS (select from a pre-defined list of operating systems that Opware SAS supports).
 - **Description:** *(Optional)* Provide a text description to identify the platform and hardware support.

- 3 Click **Next**. The Define Installation page is displayed, as Figure 3-4 shows.

Figure 3-4: Define Installation Page in the Prepare Operating System Wizard



- 4 Define the installation by providing the following information:
- **installation Media:** (*Required*) Specify the MRL for the Unix OS (select one MRL from the pre-defined drop-down list of available MRLs).
See “OS Media Management” on page 139 for more information on this topic.
 - **Build Customization Script:** (*Optional*) Click Select to choose a script to use for this installation profile from the popup window that appears. (Customization scripts that you have created will appear in the popup window after you upload them through the SAS Web Client, see “Using Build Customization Scripts” on page 171.)
The way you can customize the build process is specific to each build script. You must follow the requirements for build customization scripts to use this feature.

- **Configuration File:** *(Required)* Specify a JumpStart profile, Kickstart configuration file, or YaST2 `autoinst.xml` file to upload for use by OS Provisioning.

The file that you upload can have any file name, however, OS Provisioning renames the file during upload with the file name required by the vendor installation program.

5 Click **Upload**.

Opware SAS creates the Unix OS installation profile and uploads the configuration file (and parses packages for Sun Solaris, Red Hat, SUSE Linux, and VMware ESX). A progress bar shows the progress of the OS preparation process.

6 Click **Close** when the upload is completed.

Hardware Signature Files for Windows

A Windows response file contains some information that is applicable to any hardware make and model. Other information in the configuration file is hardware-specific, taking into account differences between specific models of servers.

The generic part of the response file specifies how to install and configure the Windows OS. Typically, the hardware-specific part specifies hardware dependent configuration for devices such as mass storage.

Based on the hardware you expect to provision, you can upload hardware-specific files for each Windows OS installation profile. You can map a signature for that hardware to the correct hardware-specific profile. OS Provisioning selects the correct Hardware Signature file at build time based on the hardware signature of the server that is about to be provisioned.

Certain x86-processor-based hardware requires pre-installation configuration. You usually perform this configuration by running vendor-supplied utilities with certain parameters. Because the utilities are hardware specific, you can script these configuration steps by using a Hardware Signature file.

Utilities referenced by the Hardware Signature file must be accessible through the network during build time.



Using Hardware Signatures is not required for Sun Solaris or Red Hat Linux operating systems because Solaris and Linux distributions do not need to be tailored for particular hardware models.

Defining an OS Installation Profile – Windows

To use the Prepare Operating System Wizard to define a Windows OS installation profile, perform the following steps:

- 1 Start the OS installation profile window from one of the following locations:
 - SAS *Web Client* Home Page: click the Prepare OS link in the **Tasks** panel. Or, from the **Navigation** pane, click Software ► Operating Systems. The Operating Systems page appears. Click **Prepare OS**.
 - SAS Client: from the **Navigation** pane, select Library ► OS Installation Profiles. Select an OS, then from the **Actions** menu, select **Create New**.(If asked to log into the SAS Web Client, enter your user name and password, and click **Log In**.)

The Describe OS page appears, see Figure 3-5.

Figure 3-5: Describe OS Page in the Prepare Operating System

The screenshot shows a web interface titled "Prepare Operating System" with a blue header. On the left is a vertical navigation pane with four steps: 1 Describe OS (highlighted in green), 2 Define Installation, 3 Upload File, and 4 Review Packages. The main content area is titled "Describe OS" and contains the instruction "Enter the following information to describe the operating system." Below this are four input fields: "Name:" (a text box), "Description:" (a text area with scrollbars), "Customer:" (a dropdown menu with "Customer Independent" selected), and "OS Version:" (a dropdown menu with "OS Independent" selected). At the bottom of the page are three buttons: "Cancel", "< Previous", and "Next >".

- 2 Describe the OS by specifying the following information:
 - **Name:** *(Required)* Specify the display name for the Windows OS.
 - **Customer:** *(Required)* Associate the Windows OS with a specific customer; to set up the OS for use by all customers, select "Customer Independent".
 - **OS Version:** *(Required)* Specify the version of the Windows OS (selected from the pre-defined list of the operating systems that Opware SAS supports).
 - **Description:** *(Optional)* Provide a text description to identify the platform and hardware support.

- 3 Click **Next**. The Define Installation page appears, see Figure 3-6.

Figure 3-6: Define Installation Page in the Prepare Operating System Wizard

The screenshot shows a web browser window titled "https://192.168.201.3 - Prepare Operating System - Mozilla Firefox". The main content area is titled "Prepare Operating System" and "Define Installation". A sidebar on the left shows a progress indicator with four steps: 1 Describe OS, 2 Define Installation (highlighted), 3 Upload File, and 4 Review Packages. The main content area has the heading "Define Installation" and the instruction "Specify the installation media, response file and optional pre-installation options." Below this are four sections: "Installation Media" with a dropdown menu set to "Windows 2003"; "Installation Options" with radio buttons for "DOS" (selected) and "WINPE"; "Build Customization" with a "None" label and "Select..." and "Delete" buttons; and "Response File" with a "None" label and a "Select..." button. At the bottom of the form are "Cancel", "< Previous", and "Upload" buttons. The status bar at the bottom shows "Done" on the left and "192.168.201.3" on the right.

- 4 Define the installation by providing the following information:
- **OS Media:** (*Required*) Specify the MRL for the Windows OS (select one MRL from the pre-defined drop-down list of available MRLs that you have already defined). See “OS Media Management” on page 139 for more information on this topic.
 - **Installation Options:** (*Required*) Choose the type of pre-installation environment (DOS or WinPE) to use when you install the Windows OS.
Your selection determines which customization script options you can use. Selecting DOS allows you to use all DOS options and some WinPE options. Selecting WinPE limits you to WinPE supported options.

When a server is booted with the WinPE pre-installation environment, it will appear in the Server Pool in the SAS Web Client and in the Unprovisioned Servers list in the SAS Client. If you select WINPE, you can set the following parameters:

- **Custom Disk Partitioning:** The script you provide is passed to the Microsoft diskpart.exe utility and is used during OS installation. Refer to the Microsoft Windows product documentation for more information.
- **Custom Disk Formatting:** This script is executed directly onto the hard drive during OS installation.
- **Install Drive:** Indicates the drive letter to install the Windows OS on.

If you do not enter any settings in these fields, the default values used will be as shown in Figure 3-7.

Figure 3-7: Default Values Used for WinPE Installation Options in OS Installation Profile

Installation Options	
Select Installation Type: <input type="radio"/> DOS <input checked="" type="radio"/> WINPE	
Custom Disk Partitioning:	<pre>rescan select disk 0 clean create partition primary active assign letter=C</pre>
Custom Disk Formatting:	<pre>format.com C: /FS:NTFS /Q /Y /V:</pre>
Install Drive:	<input type="text" value="C"/>

- **Build Customization:** (Optional) Select a build script to customize the way the build process operates for the Windows OS.

You can customize the build process specifically for each pre-installation environment. You must follow the requirements for build customization scripts to use this feature. Scripts appear in the popup window for your selection after you upload them through the SAS Web Client.

Click **Select** to choose a file from the popup window.

See “Build Customization Scripts” on page 171 for more information.

- **Response File:** (*Required*) Select a Windows response file to upload into the OS installation profile. This can be an `unattend.txt` for unattended Windows installations or a `sysprep.inf` type file for image type Windows installations. The file that you upload can have any file name, however, OS Provisioning renames the file during upload with the valid file name required by the vendor installation program.
- **Hardware Signatures:** (*Optional*) Define the list of hardware that the OS supports. Click **Add** to open the Add Hardware Signature Setting window. The **Applies To** field is pre-populated with the hardware makes and models that have been built, so that they appear in the Managed Server list. You can add multiple Hardware Signature files to a Windows OS installation profile.

5 Click **Upload**.

Opware SAS creates the OS installation profile and uploads the configuration file (and examines any packages). A progress bar shows the progress of the OS preparation process.

6 Click **Close** when the process is complete.

Modifying Existing OS Installation Profiles

You can edit an OS Installation Profile by:

- Changing the properties for the OS, for example which customer(s) can use the OS Installation Profile to provision servers.
- Modifying the way that the OS is installed on servers by changing the configuration file or customizing the way the build process works for that OS Installation Profile.
- Adding custom attributes to the OS Installation Profile to override default values in the build process. You can add custom attributes from the SAS Web Client or from the SAS Client. (See “Default Custom Attribute Values” on page 190. For information on how to set custom attributes for software policies, see “Setting Custom Attributes For Software Policies” on page 60.
- Specifying custom disk partitioning and custom drive formatting (For Windows servers booted with WinPE).
- Setting up configuration tracking for an OS Installation Profile.

See Opware[®] SAS User’s Guide: Server Automation for information on how to set a configuration tracking policy for the OS installation profile.

Changing the OS Installation Profile Properties

To change the properties for an OS installation profile:

- 1** From the **Navigation** pane, click Software ► Operating Systems. The **Operating Systems** page appears.
- 2** Click the name of the OS that you want to edit. The **Edit Operating System** page appears.
- 3** Select the **Properties** tab (see Figure 3-8). You can modify the following settings:
 - **Name:** Sets the display name for the OS.
 - **Description:** Provides a text description of the OS.
 - **Customer:** Associates the OS with a specific customer.

If you have OS Sequence client permissions, you can change the Name and Description of OS Installation Profile in the SAS client.

Note that, you cannot change the customer association for an OS installation profile.

Figure 3-8: Properties Tab for an OS Installation Profile in the SAS Web Client

Properties	Installation	Packages 0	Custom Attributes 0	Servers 0	Config Tracking	History
Name:	<input type="text" value="Windows"/>					
Description:	<input type="text"/>					
Customer:	<input type="text" value="Customer Independent"/>					
OS Version:	Windows 2003					
Packages:	0					
Last Modified:	Tue Apr 26 18:58:51 2005					
ID:	40070004					
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>					

- 4** Click **Save**.

Modifying How an OS Is Installed on a Server – Unix

To modify the way an OS is installed on Unix servers:

- 1** From the **Navigation** pane in the SAS Web Client, click Software ► Operating Systems. The **Operating Systems** page appears.

- 2** Click the name of the Unix OS that you want to edit. The **Edit Operating System** page appears.
- 3** Select the **Installation** tab.
- 4** Modify the following settings:
 - **Installation Media:** *(Required)* Set the MRL for the Unix OS (select one MRL from the pre-populated drop-down list).
See “OS Media Management” on page 139 for more information on this topic.
 - **Build Customization Script:** *(Optional)* Customize the way the build process operates for that Unix OS (select a file from the popup window).
The way you can customize the build process is specific to each build script. You must follow the requirements for build customization scripts to use this feature. Scripts appear in the popup window after you upload them through the SAS Web Client.
See “Build Customization Scripts” on page 171 for more information.
 - **Configuration File:** *(Required)* Specify a JumpStart profile, Kickstart configuration file, or YaST2 `autoinst.xml` file to upload for use by OS Provisioning.
The file that you upload can have any file name, however, OS Provisioning renames the file during upload with a valid file name required by the vendor installation program.
- 5** Click **Save**.

Modifying How an OS Is Installed on a Server - Windows

Perform the following steps to modify the way an OS is installed on Windows servers:

- 1** From the **Navigation** pane, click Software ► Operating Systems. The **Operating Systems** page appears.
- 2** Click the name of the OS that you want to edit. The **Edit Operating System** page appears.

- 3** Select the **Installation** tab. The installation resources defined for the OS installation profile appear, as Figure 3-9 shows.

Figure 3-9: Installation Tab for an OS Installation Profile in the SAS Web Client

[Return to Os Installation Profiles](#)

Properties	Installation	Packages 0	Custom Attributes 0	Servers 0	Config Tracking	History
Installation Media						
Windows 2003 x64		<input type="button" value="Select..."/>				
Installation Options						
Install Type:	WINPE	<input type="button" value="Save Install Options"/>				
Custom Disk Partitioning:	<pre>rescan select disk 0 clean create partition primary active assign letter=C</pre>					
Custom Disk Formatting:	<pre>format.com C: /FS:NTFS /Q /Y /V:</pre>					
Install Drive:	C					
Build Customization						
None		<input type="button" value="Select..."/>				
Response File						
unattend.txt		<input type="button" value="Upload..."/>				
Hardware Signatures						
<input type="button" value="Add..."/>						

- 4** You can modify the following settings:
- **Installation Media:** Set the MRL for the Windows OS. Click **Select** and select an OS media from the list in the popup window.
 - **Installation Options:** If you selected WINPE when you created the Windows installation profile, you can set the following custom disk partitioning parameters:
 - **Custom Disk Partitioning:** The script you provide is passed to the Microsoft diskpart.exe utility and is used during OS installation. Refer to the Microsoft Windows product documentation for more information.

If you leave this section blank, the following default values will be used:

```
rescan
select disk 0
clean
```

```

create partition primary
active
assign letter=C

```

- **Custom Disk Formatting:** This script is executed directly onto the hard drive during OS installation. If you leave this section blank, the default values used will be as follows:

```
format.com C: /FS:NTFS /Q /Y /V:
```

- **Install Drive:** Indicate which drive letter to install the Windows OS on. The default drive letter used is C.
- **Build Customization Script:** Customizes the way the build process operates for that OS. Click **Select** and select a build customization package from the list in the popup window.

Scripts appear in the popup window after you upload them through the SAS Web Client.

- **Configuration File:** Indicates the Windows response file to upload for use by OS Provisioning. Click **Upload** and enter the file name or browse to the file.

The file that you upload can have any file name. However, OS Provisioning renames the file with the correct file name for use by the vendor installation program.

- **Hardware Signatures for Windows only:** Defines the list of hardware that the OS supports. Click **Add** and select the hardware signature that you want to include in the OS installation profile.

Hardware signatures appear in the list box after a server with that selected make and model are successfully built, so that it appears in the Managed Server list.

5 Click **Save**.

Modifying the OS Installation Profile Packages

With the release of Opware SAS 6.5.1, you should add packages to an OS installation profile using software policies attached to OS sequences. This is because Opware SAS 6.1 and later no longer attempts to automatically calculate the list of packages to attach to the OS Installation Profile.

If you have upgraded from earlier releases, your existing OS Installation Profiles for Solaris and/or Linux will already have a list of packages attached. However, if you need to upload a new configuration file (kickstart or jumpstart profile) with a different set of packages, you must create a new profile using the Prepare OS Wizard.

Note also, that when you provision OS sequences that you migrated from SAS 5.x via the Run OS Sequence wizard, the OS Installation Profile packages are no longer remediated. If you have manually attached packages additional to the package list that was automatically generated when the profile was uploaded to the OS Installation Profile, provisioning servers with an OS Sequence referencing that OS Installation Profile will not install these extra packages. To insure that these packages are installed during provisioning, you must add them to a Software Policy, attach that policy to the OS Sequence, and enable remediation.

See “Creating an OS Installation Profile” and “Creating an OS Sequence” in the *Opware® SAS 6.5 User's Guide: Application Automation* for more information.

The method described in this section is provided for those using versions of Opware SAS prior to 6.1

Perform the following steps to modify the packages that an OS installation profile installs:

- 1** From the **Navigation** pane, click Software ► Operating Systems. The **Operating Systems** page appears.
- 2** Click the display name of the OS that you want to edit. The **Edit Operating System** page appears.
- 3** Select the **Packages** tab. The list of packages that the OS installation profile installs appears, as Figure 3-10 shows.

Figure 3-10: Packages Tab for an OS Installation Profile in the SAS Web Client

Properties	Installation	Packages 117	Custom Attributes 0	Servers 0	Config Tracking	History
The following Packages are Directly Attached to this Node Edit Package Attachments						
Name	Type	Description				
vim-common-6.0-0.27.i386	RPM	The common files needed by any version of the VIM editor.				
tar-1.13.19-4.i386	RPM	A GNU file archiving program.				
gettext-0.10.35-31.i386	RPM	GNU libraries and utilities for producing multi-lingual messages.				
sh-utils-2.0-13.i386	RPM	A set of GNU utilities commonly used in shell scripts.				
mount-2.10r-5.i386	RPM	Programs for mounting and unmounting filesystems.				

- 4** Click **Edit Packages**. The Software Directly Attached page appears.
- 5** To add a package for installation, click **Add Software** and specify or search for the package that you want to add to the list.
- 6** To remove packages, select them in the list and click **Remove Software**. The packages are deleted from the list in the page but are not actually removed from the OS installation profile until you click **Save Edits**.
- 7** To change the order in which the packages are installed on servers, select the package that you want installed in a different order and click the up or down arrows.
- 8** Click **Save Edits**.

Viewing Change History for an OS Installation Profile

By default, OS Provisioning maintains information about the changes to OS installation profiles for 180 days.

The following actions create an entry in the History of an OS installation profile:

- The customer association is changed for the OS installation profile.
- A server uses the OS installation profile to install an OS.
- Packages are added to or removed from the Package List in the OS installation profile.

You can view the history of changes to an OS Installation profile in the SAS Web Client and in the SAS Client.

To view the history of changes to an OS installation profile in the SAS Web Client, perform the following steps:

- 1** From the **Navigation** pane, click Software ► Operating Systems. The **Operating Systems** page appears.
- 2** Click on the name of the OS to review the history of its changes. The **Edit Operating System** window appears.

- 3 Select the **History** tab. The list of events and changes appears, as Figure 3-11 shows.

Figure 3-11: History Tab for an OS Installation Profile in the SAS Web Client

[Return to Operating Systems](#)

Properties	Installation	Packages 1278	Custom Attributes 0	Servers 0	Config Tracking	History
HISTORY FOR: Red Hat Linux 7.3 / 7.3 for precision 360s by mwp						
						Show Last: Week Two Weeks Month Quarter
Event Description	Modified By	Date Modified				
Removed package id 24610028 from node 7.3 for precision 360s by mwp	mpound	Wed May 18 18:20:59 2005				
Removed package id 23230029 from node 7.3 for precision 360s by mwp	mpound	Wed May 18 18:20:58 2005				
Removed package id 24560029 from node 7.3 for precision 360s by mwp	mpound	Wed May 18 18:20:00 2005				
Removed package id 25170028 from node 7.3 for precision 360s by mwp	mpound	Wed May 18 18:20:00 2005				

To view the history of changes to an OS installation profile in the SAS Client, perform the following steps:

- 1 Launch the SAS Client using one of the following methods:
 - From the **Power Tools** section of the SAS Web Client home page
 - From **Start ► All Programs ► SAS Client**
- 2 From the **Navigation** pane, select Library ► OS Installation Profiles.
- 3 Browse an OS installation profile and open it. The **OS Installation Profile** window opens.
- 4 From the **Navigation** pane, select History. The **Content** pane shows the history of changes to the OS Installation Profile.

Deleting an OS Installation Profile



If a server is currently using an OS installation profile or an OS installation profile is included in a template, you cannot delete it.

To delete an OS installation profile, perform the following steps:

- 1 From the **Navigation** pane, click Software ► Operating Systems. The **Operating Systems** page appears.
- 2 Select the OS that you want to delete.

- 3 Click **Delete**. (If a server has used the OS installation profile or the OS installation profile is included in a template, a warning message appears.)

The list of OS installation profiles re-appears.

Build Customization Scripts

This section discusses the following topics:

- Using Build Customization Scripts
- Before Creating a Sun Solaris Build Customization Script
- Solaris Build Customization Script
- Solaris Provisioning from a Boot Server on a Redhat Linux Server
- Requirements for Solaris Build Customization Scripts
- Sample Solaris Build Customization Script
- Linux Build Process
- Linux Build Customization Scripts
- Requirements for Linux Build Customization Scripts
- VMware ESX Build Process
- VMware ESX Build Customization Scripts
- Windows Build Process (DOS Boot Image)
- Windows Build Customization Scripts (DOS Boot Image)

Using Build Customization Scripts

You can use OS-specific build scripts to control the way each OS is provisioned. Build scripts allow you to manage each OS installation from the network connection to Opware Agent installation.

OS provisioning build scripts provide hooks into the build process that allow you to modify OS installations at specific points. These hooks call a single build customization script at the appropriate time in the OS installation process.

Because each build script is specific to the OS it installs, build customization and installation vary by OS. Before you can use a build customization script as part of an OS installation profile, you need to create the build customization script and import it into the Opware SAS Client.

To import a build customization script into the SAS Client, perform these tasks:

- 1** From the **Navigation** pane, select Library ► Packages and then select an operating system.
- 2** From the **Actions** menu, select **Import OS Utilities**.
- 3** In the Import OS Utilities window, click **Browse** to select the build customization script. Note that, dependent on the OS, the customization script filename is expected to follow certain conventions (for example, the Solaris script must be a Bourne shell script and must be named `run`). See the section for your OS below for information about these conventions.
- 4** From the **Customer** list, select a customer to associate with the build customization script.
- 5** From the **Platforms** list, select an operating system platform to associate with the build customization script.
- 6** Click **Import**.

Later, When you are preparing an OS installation profile you will have the opportunity to select a build customization script to associate with the profile. Build customization scripts that you have imported as described above will appear in a list when you click **Select**.

See “Defining an OS Installation Profile – Unix” on page 155 or “” on page 159 for more information.

Before Creating a Sun Solaris Build Customization Script

It is important to understand the Solaris build process before you include a build customization script for a Solaris installation profile. Table 3-5 details the exact steps that occur when you provision an installation client with Solaris.

A user initiates the build process with Steps 1 and 5. The rest of the build process steps occur automatically in OS Provisioning.

Table 3-5: Sun Solaris Build Process

PHASE	BUILD PROCESS STEPS
Pre-installation	<ol style="list-style-type: none"> <li data-bbox="545 508 1332 629">1 A user boots the installation client over the network by entering the following command in a console attached to the server: <code>boot net:dhcp - install</code> <li data-bbox="545 639 1332 794">2 The installation client boots from the network by using a Solaris 10 JumpStart miniroot (included as part of OS Provisioning), eventually running a JumpStart <code>begin</code> script. The <code>begin</code> script is used to start the Opware OS Build Agent. <li data-bbox="545 813 1332 890">3 The OS Build Agent registers with the Opware OS Build Manager. <li data-bbox="545 909 1332 1054">4 The Solaris <code>build</code> script probes the hardware configuration of the installation client and registers it with Opware SAS. The installation client then appears in the Server Pool list in the SAS Web Client.

Table 3-5: Sun Solaris Build Process (continued)

PHASE	BUILD PROCESS STEPS
Phase One	<p>5 In the SAS Web Client, a user chooses to install an OS on an available installation client.</p> <p>6 The Solaris build script mounts the Solaris installation media indicated by the MRL in the OS installation profile that the user selected.</p> <p>7 The Solaris build script retrieves the profile associated with the selected OS installation profile and copies it to <code>\$SI_PROFILE</code>, the standard JumpStart location for dynamic JumpStart profiles.</p> <p>8 The Solaris build script executes the build customization script:</p> <pre data-bbox="602 819 1017 846">/sbin/sh run Pre-JumpStart</pre> <p>9 The Solaris build script validates the profile by using the JumpStart installer (<code>pfinstall</code>) in test mode.</p> <p>10 The Solaris build script causes the OS Build Agent to run in the background, allowing the JumpStart <code>begin</code> script to complete.</p> <p>11 The JumpStart installer <code>pfinstall</code> is invoked by the JumpStart installer script and Solaris is installed. Concurrently, the OS Build Agent monitors the installation process. Feedback is displayed in the SAS Web Client.</p> <p>12 The JumpStart installer <code>pfinstall</code> completes and runs the JumpStart <code>finish</code> script, which indicates to OS Provisioning that the OS installation is complete.</p> <p>13 The build script executes the build customization script a second time:</p> <pre data-bbox="602 1476 1033 1503">/sbin/sh run Post-JumpStart</pre> <p>14 The installation client reboots.</p>

Table 3-5: Sun Solaris Build Process (continued)

PHASE	BUILD PROCESS STEPS
Phase Two	<p>15 On entering multiuser mode, the OS Build Agent is invoked and it contacts the OS Build Manager.</p> <p>16 The Solaris build script executes the build customization script: <code>/sbin/sh run Pre-Agent</code></p> <p>17 The Solaris build script installs the Opsware Agent.</p> <p>18 The Solaris build script executes the build customization script: <code>/sbin/sh run Post-Agent</code></p> <p>19 The Solaris build script exits and Phase Two finishes.</p> <p>20 OS Provisioning takes over, causing a remediation of the selected software to be installed onto the installation client.</p>

See the *Opsware® SAS Content Migration Guide* for more information on how remediation works to install software on servers.

Solaris Provisioning from a Boot Server on a Redhat Linux Server

If you need to provision a Solaris server and the Opsware Boot Server is on a Redhat server, you must disable NFS v3 on the Boot Server. (If the Boot Server is on a Solaris server, do not perform this action.)

Disabling NFS v3

To disable NFS v3, perform the following steps

- 1** On the Boot Server host, create the following file:
`/etc/sysconfig/nfs`
- 2** In the newly created `nfs` file, add the following line:
`MOUNTD_NFS_V3=no`
- 3** Restart NFS:
`/etc/init.d/nfs stop`
`/etc/init.d/nfs start`

Solaris Build Customization Script

You can customize a Solaris installation at multiple points using a build customization script. The following list shows these points:

- **Pre-JumpStart:** A pre-installation hook for the first stage.

During Phase One, the build customization script runs in the JumpStart environment. The script can use all the standard JumpStart environment variables, such as `SI_PROFILE`. All the environment variables associated with the standard JumpStart probe keywords and values are set (for example, `SI_DISKLIST`, `SI_HOSTADDRESS`, and `SI_MEMSIZE`).

When the `run` script is invoked at the Pre-JumpStart point, it can perform any actions that a JumpStart `begin` script would perform. For example, the script could modify the downloaded profile before the OS installation begins. At this point, the Solaris profile is downloaded from OS Provisioning, but the profile has not been passed to the JumpStart server.

For the complete list of the environment variables, see the *Solaris 9 Installation Guide*.

- **Post-JumpStart:** A post-installation hook for the first stage.

When the `run` script is invoked at the Post-JumpStart point, it can perform any actions that a JumpStart `finish` script would perform. One example would be to set custom `eeprom` settings. The installation client's file systems are available for modification at this point and are mounted on the `/a` partition for the `finish` script environment.

- **Pre-Agent:** A pre-installation hook for the second stage.
- **Post-Agent:** A post-installation hook for the second stage.

During Phase Two, the `run` script is executed after the installation client has rebooted. This is the point when the system is up and running in multi-user mode with most services started.

The last 4K of output produced by the build customization script (`stdout` and `stderr`) appears in the SAS Web Client output details for the OS.

Requirements for Solaris Build Customization Scripts

To use a build customization script for Solaris, you must meet the following requirements:

- You must create the script as a Bourne shell script and name it `run`.

- You must include the `run` script in an archive file in `tar.Z` format and include the script at the top level of the archive. During OS provisioning, the `tar.Z` archive is unpacked on the installation client and the script is processed by `/sbin/sh`.
- You must be sure that the `run` script is unpacked in its own directory with the other files in the archive. This directory serves as the current working directory when the `run` script is invoked. Based on this fact, correctly refer to the other files in the archive. For example, unpacking and invoking the `run` script follows this general process:

```
mkdir /var/tmp/inst_hook
cd /var/tmp/inst_hook
zcat hook.tar.Z | tar xf -
/sbin/sh run <stage>
```

- You must create a script that cannot cause the installation client to drop its network connection (for example, do not use the script to reboot the installation client or reconfigure the active network interface). If the installation client drops its network connection, the OS provisioning process will fail.
- You must create the `run` script so that it exits normally. If the script exits with a non-zero value, the OS provisioning process will end. However, the JumpStart process will continue when a pre-installation hook fails (exits with a non-zero value). When creating the `run` script, you should ensure that the JumpStart process does not continue when a pre-installation hook fails.

The `run` script should not take an exceptionally long time to complete, otherwise the OS provisioning process might time out.

Sample Solaris Build Customization Script

```
#!/sbin/sh
pre_jumpstart() {
    #
    # strip any partitioning information out of profile, and
    # replace it with keywords to use default partitioning, but
    # to size swap equal to the amount of physical RAM
    #
    cat $SI_PROFILE | grep -v partitioning | grep -v filesys > /tmp/
profile.$$
    echo "partitioning default" >> /tmp/profile.$$
    echo "filesys any $SI_MEMSIZE swap" >> /tmp/profile.$$
    cp /tmp/profile.$$ $SI_PROFILE
    rm -f /tmp/profile.$$
}
post_jumpstart() {
```

```
#
# set local-mac-address eeprom setting
#
eeprom 'local-mac-address?=true'
}
pre_agent() {
    : # do nothing
}
post_agent() {
    : # do nothing
}
case "$1" in
    Pre-JumpStart) pre_jumpstart ;;
    Post-JumpStart) post_jumpstart ;;
    Pre-Agent)     pre_agent ;;
    Post-Agent)    post_agent ;;
esac
```

Linux Build Process

It is important to understand the Linux build process before you include a build customization script in a Linux OS installation profile. Table 3-6 describes the exact steps that occur when you provision an installation client with Red Hat or SUSE Linux.

A user initiates the build process with Steps 1 and 6 and the rest of the build process steps happen automatically in OS Provisioning.



The build process for VMware ESX follows the same process as the Linux build process.

Table 3-6: Linux Build Process

PHASE	BUILD PROCESS STEPS
Pre-installation	<ol style="list-style-type: none"> 1 A user boots the installation client from PXE or the Linux Boot CD ROM. 2 The installation client loads a custom Red Hat AS 3.0 boot image and mounts the second stage image specified by the kernel parameters. 3 Anaconda is replaced by a custom Opsware script that is used to invoke the OS Build Agent. 4 The OS Build Agent registers with the Opsware Build Manager. 5 The Linux build script probes the hardware configuration of the installation client and registers it with Opsware SAS, causing the installation client to appear in the Server Pool list in the SAS Web Client.

Table 3-6: Linux Build Process (continued)

PHASE	BUILD PROCESS STEPS
Phase One	<p>6 In the SAS Web Client, a user selects the target version of Linux to install on the installation client.</p> <p>7 The Linux build script creates a small partition at the beginning of the disk and copies the target boot image from the Boot Server to this partition.</p> <p>8 The Linux build script copies GRUB onto the partition and installs it into the MBR.</p> <p>9 The Linux build script configures GRUB to boot this partition, and kernel arguments are set to do an NFS installation on the location indicated by the MRL.</p> <p>10 If the Custom Attribute <code>kernel_arguments</code> is set for the OS installation profile, these kernel arguments are appended.</p> <p>11 The OS Build Agent exits and the server reboots.</p>
Phase Two	<p>12 The target boot image loads and runs the OS Build Agent.</p> <p>13 The Linux build script verifies that the media indicated by the MRL is the same version as the boot image under which it is running.</p> <p>14 The Linux build script writes the configuration file defined by the MRL to the disk.</p> <p>15 If it exists, the Linux build script runs the build customization script.</p> <p>16 The Linux build script runs in the background. The OS Build Agent and Anaconda starts. The Linux installation starts normally by using the configuration file written to the disk. Concurrently, the OS Build Agent monitors the installation process providing feedback, which is displayed in the SAS Web Client.</p> <p>17 After all packages have been installed, the OS Build Agent copies the Opware Agent Installer and the OS Build Agent to the server and sets up an <code>init</code> script to start the OS Build Agent after the reboot.</p> <p>18 When the OS installation completes, Anaconda reboots the installation client, which will boot from the newly installed OS.</p>

Table 3-6: Linux Build Process (continued)

PHASE	BUILD PROCESS STEPS
Phase Three	<p>19 On entering multi-user mode, the OS Build Agent is invoked and contacts the OS Build Manager.</p> <p>20 The Linux build script installs the Opsware Agent.</p> <p>21 The Linux build script exits.</p> <p>The OS installation section of provisioning is complete.</p>

Linux Build Customization Scripts

The Linux build script runs a single installation hook that gives you the ability to customize the Linux build process before Anaconda loads.

The installation hook is run in a RAM disk right before the installation program runs, but after the network has been brought up.

Requirements for Linux Build Customization Scripts

To use a build customization script for Linux, you must meet the following requirements:

- You must create an executable script and name it `run`.
- You must include the `run` script in an archive file in `tar.gz` format and include the script at the top level of the archive. During OS provisioning, the `tar.gz` archive is unpacked on the installation client and the script is executed.
- You must unpack the `run` script in its own directory with the other files in the archive. This directory serves as the current working directory when the `run` script is invoked. Based on this fact, correctly refer to the other files in the archive. For example, unpacking and invoking the `run` script follows this general process:

```
mkdir /tmp/installhook
cd /tmp/installhook
tar -xzf hook.tgz
./run 2>&1
```

- You must ensure that the `run` script does not take an exceptionally long time to complete, otherwise the OS provisioning process might time out.
- You must ensure that the `run` script exits normally. If the script exits with a non-zero value, the OS provisioning process will end.
- You must ensure that the `run` script has execute permissions to function properly.

VMware ESX Build Process

The VMware ESX build process follows the same general steps as the Linux build process.

The main difference between the VMware ESX and Linux is that VMware ESX ships by default with an `iptables` firewall that will block communication between the core and the mini-agent and agent. In order for the mini-agent to work correctly, build scripts add firewall rules and these rules allow the traffic needed for the mini-agent to function. The agent for VMware ESX is also enhanced to manage the necessary allow rules, which enables the flow of communication between the Opware agent and core.

The rest of the VMware ESX build process follows the same process as the Linux build process. For more information, see “Linux Build Process” on page 179.

VMware ESX Build Customization Scripts

The VMware ESX build script runs a single installation hook that gives you the ability to customize the VMware ESX build process before Anaconda loads.

The installation hook is run in a RAM disk right before the installation program runs, but after the network has been brought up.

Windows Build Process (DOS Boot Image)

Table 3-7 details the steps that occur when you provision an installation client with Windows.

A user initiates the build process with Steps 1 and 6. The rest of the build process steps happen automatically in OS Provisioning.

Table 3-7: Microsoft Windows Build Process (DOS Boot Image)

PHASE	BUILD PROCESS STEPS
Pre-installation	<ol style="list-style-type: none"> <li data-bbox="529 658 1333 730">1 A user boots an installation client over the network by using a PXE network bootstrap program or by using the Windows Boot Image. <li data-bbox="529 755 1333 826">2 The user selects a Windows boot type from the menu on the console for the PXE network bootstrap program. <li data-bbox="529 852 1333 923">3 PXE boots the Windows Opsware OS Build Agent over the network. <li data-bbox="529 948 1333 1020">4 The Opsware OS Build Agent prompts the user to create a FAT boot partition on which to install Windows. <li data-bbox="529 1045 1333 1213">5 The Opsware OS Build Agent collects pertinent hardware information and registers the information with Opsware SAS. The server is ready to be provisioned and is available for selection from the Server Pool in the SAS Web Client.
Phase One	<ol style="list-style-type: none"> <li data-bbox="529 1238 1333 1348">6 The user selects a Windows server from the Server Pool list in the SAS Web Client and assigns a Windows OS installation profile or a Windows template to the server. <li data-bbox="529 1373 1333 1445">7 If you specified a build customization script in the OS installation profile, the Windows build script runs the Pre-Copy hook. <li data-bbox="529 1470 1333 1541">8 The Windows build script mounts the Windows installation media as indicated by the Media Resource Location (MRL). <li data-bbox="529 1566 1333 1599">9 The Windows build script initiates a Windows unattended setup. <li data-bbox="529 1624 1333 1696">10 The Windows build script waits for a Windows unattended setup to complete and Windows to boot for the first time.

Table 3-7: Microsoft Windows Build Process (continued)(DOS Boot Image)

PHASE	BUILD PROCESS STEPS
Phase Two	<p>11 Windows boots for the first time.</p> <p>12 If you specified a build customization script in the OS installation profile, the Windows build script runs the Pre-Agent hook.</p> <p>13 If a build customization script was specified in the OS installation profile, it is executed by the Windows build script.</p> <p>14 The Windows build script installs the Opware Agent.</p> <p>The Windows build script exits and Phase Two is complete.</p>

Windows Build Process (WinPE Boot Image)

Table 3-7 details the steps that occur when you provision an installation client with Windows WinPE.

A user initiates the build process with Steps 1 and 6. The rest of the build process steps happen automatically in OS Provisioning.

Table 3-8: Microsoft Windows Build Process (WinPE)

PHASE	BUILD PROCESS STEPS
Pre-installation	<p>1 A user boots an installation client over the network by using a PXE network bootstrap program or by using the WinPE.</p> <p>2 The user can install either WinPE x86 32 bit or WinPE x64 64 bit pre-installation environment.</p> <p>3 PXE boots the Windows Opware OS Build Agent over the network.</p> <p>Note that, when using the WinPE pre-installation environment, you will not be prompted to create a disk partition.</p> <p>4 The Opware OS Build Agent collects pertinent hardware information and registers the information with Opware SAS.</p> <p>The server is ready to be provisioned and is available for selection from the Server Pool in the SAS Web Client.</p>

Table 3-8: Microsoft Windows Build Process (continued)(WinPE)

PHASE	BUILD PROCESS STEPS
Phase One	<p>5 The user selects a Windows server from the Server Pool list in the SAS Web Client and assigns a Windows OS installation profile or a Windows template to the server.</p> <p>6 The Windows build script mounts the Windows installation media as indicated by the Media Resource Location (MRL).</p> <p>7 The Windows build script initiates a Windows unattended setup.</p> <p>8 The Windows build script waits for a Windows unattended setup to complete and Windows to boot for the first time.</p>
Phase Two	<p>9 Windows boots for the first time.</p> <p>10 If a build customization script was specified in the OS installation profile, it is executed by the Windows build script.</p> <p>11 The Windows build script installs the Opsware Agent.</p> <p>The Windows build script exits and Phase Two is complete.</p>

Legacy Build Customization Script run.bat

In previous releases of Opsware SAS, OS Provisioning supported a single hook script named `run.bat`. If you choose to use this legacy script, it will still work, but it will only call the Pre-Agent hook.

For example, if the cabinet file does NOT contain a `runphase.bat` script at the root level, but it DOES contain a `run.bat` script at the top level, it will be treated as a legacy single-hook script. It will NOT be run at the "Pre-Copy" phase. It will only be run at the Pre-Agent phase with no command line arguments.

If the cabinet file contains both `runphase.bat` and `run.bat`, it will still be treated as multi-phase and `run.bat` will be ignored.

Windows Build Customization Scripts (DOS Boot Image)

The Windows build script supports the following two installation hooks:

- **Pre-Copy:** This allows you to execute code from your Windows build customization script just before the Windows Installer (`setup.exe`) is run, while the server is still running the DOS based boot image. For example, this hook could be used to make changes to the `unattend.txt` file before `setup.exe` reads it to insert hardware-specific modifications.
- **Pre-Agent:** This allows you to execute code from your Windows build customization script after the Windows OS is installed but before the Opware Agent is installed on the server. For example, you can use the hook as an opportunity to perform common post OS-installation tasks for Windows, such as modifying the Windows Registry or applying security templates.

These installation hooks must be packaged as a Microsoft cabinet file. During the provisioning process, the cabinet file is downloaded to the server being provisioned and extracted into a private temporary directory at both the Pre-Copy and Pre-Agent phases.

OS Provisioning expects to find a file named `runphase.bat` in the top level directory of the cabinet archive. If the build customization script finds a file named `runphase.bat`, it will be executed as a multi-hook script and invoked at both phases, with the phase name as the first command line argument:

```
call runphase.bat "Pre-Copy"  
call runphase.bat "Pre-Agent"
```

Windows Build Customization Scripts (WinPE)

Windows WinPE customization scripts support the following installation hooks:

- Pre-Partition
- Pre-ShareConnect
- Pre-Copy
- Post-Copy
- Pre-Reboot
- Pre-Agent
- Post-Agent

The following conventions also apply:

- WinPE Windows build customizations must be in the form of a zip file
- There must be a `run.cmd` script in the root of the zip file. See the example `run.cmd` below.
- Hooks are unpacked in `%systemdrive%\opswba\hook` (for example, `x:\opswba\hook`)
 - Hooks are unpacked recursively and will overwrite existing files.
 - Hooks are transferred and unpacked only once during the initial phase. Subsequent runs do not require unpacking. Hooks will be transferred and unpacked again after reboots (for example, before Pre-Agent), at which point they are unpacked in `%systemdrive%\opswba\hook` (typically `c:\opswba\hook`).
- When hooks are executed, the current directory will be the root directory of the unpacked zip file.
- In order to identify which phase of the build customization is being run, the build scripts pass a single command line argument to the `run.cmd` script, matching the name of the hook phase (Pre-Copy, Post-Copy, etc.). See the example `run.cmd` below.
- Note that, the build interprets a non-zero return code from a customization (hook) phase as a fatal error. Therefore, ensure that the appropriate code is returned. In the event of a fatal error, the directory in which the build customization was unpacked will be left as is (to aid in debugging). Note that this type of error is one of the few errors during the early phases of the provisioning process from which auto-recovery is not possible.
- Any output from the build customization (hook) phase will be recorded in the build log, up to and including the OCCC. Therefore, it is important to ensure that no inappropriately sensitive information is contained in the output.
- Note that, upon completion of the last build customization hook (Post-Agent), the hook directory will be forcibly deleted along with all its contents.
- After running each hook, buildscripts look for a file called `%temp%\skipnextstep`. If this file exists, it will be deleted and the next step of the provisioning will be bypassed. The following is what is bypassed for each build customization phase if the `skipnextstep` file exists:
 - Pre-Partition
 - skips partitioning and formatting

- Pre-ShareConnect
 - skips connecting Z: to the media server share
- Pre-Copy
 - skips launching the build and monitoring it altogether
- Post-Copy
 - skips copying the Opware agent and installing the boot agent (not recommended)
- Pre-Reboot
 - skips the reboot (not recommended)
- Pre-Agent
 - skips the agent install
- Post-Agent
 - skipnextstep has no effect (the file will be deleted)

Sample run.cmd File

This section shows a sample, minimal run.cmd. This sample simply echoes to the console for each hook phase. Note that, to manually test this hook from a command shell, you should execute it by using:

```
cmd /c run.cmd
```

which mimics the build agent environment as closely as possible (and prevents an exit from the script from exiting your command shell).

```
@echo off

if x%1 == xPre-Partition (
    call :PrePartition
) else if x%1 == xPre-ShareConnect (
    call :PreShareConnect
) else if x%1 == xPre-Copy (
    call :PreCopy
) else if x%1 == xPost-Copy (
```

```
        call :PostCopy
    ) else if x%1 == xPre-Reboot (
        call :PreReboot
    ) else if x%1 == xPre-Agent (
        call :PreAgent
    ) else if x%1 == xPost-Agent (
        call :PostAgent
    )
goto :end

:PrePartition
echo We are in the Pre-Partition hook phase
exit 0

:PreShareConnect
echo We are in the Pre-ShareConnect hook phase
exit 0

:PreCopy
echo We are in the Pre-Copy hook phase
exit 0

:PostCopy
echo We are in the Post-Copy hook phase
exit 0

:PreReboot
```

```
echo We are in the Pre-Reboot hook phase
exit 0

:PreAgent
echo We are in the Pre-Agent hook phase
exit 0

:PostAgent
echo We are in the Post-Agent hook phase
exit 0

:end
```

Default Custom Attribute Values

This section discusses the following topics:

- Custom Attributes for Sun Solaris
- Custom Attributes for Linux or VMware ESX
- Custom Attributes for Microsoft Windows
- Adding Custom Attributes to OS Installation Profile (SAS Web Client)
- Adding Custom Attributes to OS Installation Profile (SAS Client)

In addition to the customization provided by using build customization scripts, each build script uses custom attributes.

The SAS Web Client and SAS Client provide a data management function by allowing users to set custom attributes for servers. These custom attributes include setting miscellaneous parameters and named data values. Users can write scripts that use these parameters and data values when performing a variety of functions, including network and server configuration, notifications, and CRON script configuration.

See "Setting Custom Attributes For Software Policies" on page 60. custom attributes.

For OS provisioning, Opware SAS uses custom attributes to pass specific information to each build script to configure the installation process.

You can edit an OS installation profile to override the default values used by the build process. You override these default values by setting custom attributes for the OS installation profile.

See “Adding Custom Attributes to OS Installation Profile (SAS Web Client)” and “Adding Custom Attributes to OS Installation Profile (SAS Client)” on page 196 for specific steps required to set custom attributes for an OS installation profile.

Custom Attributes for Sun Solaris

The build script for Solaris OS provisioning uses a number of custom attributes. Several of these custom attributes correlate with an equivalent setting that would be defined normally by a Solaris `sysidcfg` file.

You cannot modify the `sysidcfg` file that OS Provisioning uses. However, you can override specific values specified in the default `sysidcfg` file. You can set custom attributes for a Solaris OS installation profile in the SAS Web Client.

The custom attributes correspond to the equivalent keywords in the `sysidcfg` file. See Table 3-9.

Table 3-9: Sun Solaris Custom Attributes

KEYWORD	DESCRIPTION
<code>archive_location</code>	NFS path to a Flash Archive (flar) to use instead of OS media.
<code>boot_options</code>	Solaris kernel parameters. These can be found in <code>/boot/grub/menu.lst</code> on X86, as EEPROM values on SPARC machine systems, or <code>bootenv.rc</code> . Values will vary.
<code>reboot_command</code>	The agent uses this command to issue a reboot. For example, "init 6", "reboot", variations of "shutdown", etc.
<code>root_password</code>	Sets the encrypted value for the password on an installation client. One way to obtain an encrypted value is by using <code>/etc/shadow</code> . If a value is not set, the system will not have a root password.

Table 3-9: Sun Solaris Custom Attributes (continued)

KEYWORD	DESCRIPTION
<code>timezone</code>	<p>Sets the time zone for the configuration of the installation client (sets <code>TZ</code> in <code>/etc/default/init</code>). The directories and files in the directory <code>/usr/share/lib/zoneinfo</code> provide the valid time zone values.</p> <p>By default, the time zone value is UTC.</p> <p>For example, the time zone value for Pacific Standard Time in the United States is <code>US/Pacific</code>. You can also specify any valid Olson time zone.</p>
<code>system_locale</code>	<p>Sets the language for the configuration of the installation client (sets <code>LANG</code> in <code>/etc/default/init</code>). Valid locale values are installed in <code>/usr/lib/locale</code>. If you set this attribute, you should also use the <code>locale</code> keyword in the operating system profile so that the appropriate locale is installed.</p> <p>By default, the value for this keyword is <code>system_local=C</code>.</p>
<code>required_patches</code>	<p>This keyword is reserved by the Solaris build script. Using it might cause the installation process to fail.</p> <p>To specify required patches, include them with the OS installation profile.</p>
<code>nfsv4_domain</code>	<p>Sets the system's default NFS version 4 domain name.</p> <p>If this value is not set, OS Provisioning suppresses the prompt to confirm the NFS version 4 domain name when the server starts the first time.</p>

Custom Attributes for Linux or VMware ESX

You can use custom attributes to specify additional arguments to the kernel under which the installation is running.

Setting a custom attribute for the OS installation profile requires that you edit the OS installation profile and select the Custom Attributes tab. The custom attribute must have the name, `kernel_arguments`.

The kernel arguments are separated by spaces (like they are when you type them after the boot prompt for the CD-ROM or DVD). For example:


```
name=value jones=barbi
```

To have the kernel arguments persist after the base OS is installed, you must set them in the uploaded configuration file. Setting kernel arguments by using custom attributes only allows you to create a completely automated installation (as if you were installing the OS from CD-ROM or DVD).

Table 3-10: Linux or VMware ESX Custom Attributes

KEYWORD	DESCRIPTION
boot_kernel	Values: "RH", "RH30", "RH40", "RH50", "RHIA"
kernel_arguments	Values: "noapci", "root=LABEL=/", "quiet", "splash"
ksdevice	Values: "eth0", "eth1", etc

Custom Attributes for Microsoft Windows

For a Windows OS installation profile, you can set various Windows OS custom attributes that allow you to replace or insert values inside the `unattend.txt` file during the OS installation process. At install-time, the resolved value of the custom attribute will be inserted into `unattend.txt`.

For example, if you do not have `AdminPassword=Foo` in your `unattend.txt` file, but you do have it added as a custom attribute, OS provisioning will automatically add `AdminPassword=CustAttrValue` at install time.

For more information on how to add custom attributes, see “Adding Custom Attributes to OS Installation Profile (SAS Web Client)” on page 195 or “Adding Custom Attributes to OS Installation Profile (SAS Client)” on page 196.

Refer to Microsoft documentation for syntax and valid values. Unless otherwise noted in the table, there are no default values for these attributes if they are not set.

Table 3-11: Windows Custom Attributes for OS Provisioning

KEYWORD	CORRESPONDING UNATTEND.TXT ATTRIBUTE	NOTES
AdminPassword	[GuiUnattended] / AdminPassword	This option sets the Administrator password for the Admin account.

Table 3-11: Windows Custom Attributes for OS Provisioning (continued)

KEYWORD	CORRESPONDING UNATTEND.TXT ATTRIBUTE	NOTES
argstring	None	String value that is used to compose the command line arguments for the Opware agent installer.
auto_partition		Used by consoleless to indicate that instead of requiring interactive user confirmation before partitioning the disk, partition the disk automatically.
ComputerName	[UserData] / ComputerName	This value is not validated by Opware. This custom attribute should only be set on the Server, but Opware will not prevent you from setting the attribute anywhere. The default value is an Opware-generated random string.
imageexec	None	Command to apply legacy image-based provisioning image. This supports traditional imaging tools such as Symantec Ghost™. However, using the built in support for WIM images is strongly encouraged.
imagefile	None	Path to a server image file. This supports traditional imaging tools such as Symantec Ghos™t. However, using the built in support for WIM images is strongly encouraged.

Table 3-11: Windows Custom Attributes for OS Provisioning (continued)

KEYWORD	CORRESPONDING UNATTEND.TXT ATTRIBUTE	NOTES
imageshare	None	Share with image file to install. This supports traditional imaging tools such as Symantec Ghost™. However, using the built in support for WIM images is strongly encouraged.
ProductKey	[UserData] / ProductKey	This value is not validated by Opsware.
timeout	None	An integer value in minutes that the Windows Setup will timeout. Default is 120 minutes. If Windows setup does not complete in the specified amount of time, the OS installation will fail with a timeout error.

Adding Custom Attributes to OS Installation Profile (SAS Web Client)

Perform the following steps to add custom attributes to an OS installation profile in the SAS Web Client:

- 1** From the **Navigation** pane inside the SAS Web Client, click Software ► Operating Systems. The **Operating Systems** page appears.
- 2** Click the name of the OS that you want to edit. The **Edit Operating System** page appears.

- 3 Select the **Custom Attributes** tab. The list of custom attributes specified for the OS installation profile appears, as Figure 3-12 shows.

Figure 3-12: The Custom Attributes Tab for an OS Installation Profile in the SAS Web Client

Properties		Installation		Packages 0		Custom Attributes 0		Servers 0		Config Tracking		History	
The following custom attributes are for this node										<input type="button" value="Add Custom Attributes"/>			
Name						Value							
No custom attributes created for this node. Click the "Add" button to add new custom attribute													




If the OS installation profile contains custom attributes, the **Edit Custom Attributes** button appears on the page. Click **Edit Custom Attributes** to add new attributes and edit existing ones.

- 4 Click **Add Custom Attribute**.
- 5 Enter a name and a value for the custom attribute.
- 6 Click **Save**. The list of custom attributes set for the OS installation profile reappears. The new custom attribute is added to the list.

Adding Custom Attributes to OS Installation Profile (SAS Client)

To add custom attributes to an OS installation profile in the SAS Client, perform the following steps:

- 1 Launch the SAS Client using one of the following methods:
 - *SAS Web Client home page*: From the Power Tools section
 - *SAS Web Client Menu*: From **Start** ► **All Programs** ► **SAS Client**
- 2 From inside the SAS Web Client, from the **Navigation** pane, select Library ► OS Installation Profiles. Ensure that you have selected the **By Type** tab.
- 3 Browse to an OS installation profile and open it. The **OS Installation Profile** window opens.
- 4 In the OS Installation Profile window, select **Custom Attributes** from the **Views** pane.
- 5 In the **Content** pane, click **Add** to add a custom attribute.

- 6** In the **Name** column, double-click a cell in the table and type a custom attribute name.
- 7** In the **Value** column, double-click a cell in the table and type a custom attribute value. If you would like to enter a longer value, click  to open a window that allows you to enter a longer value.
- 8** To delete a custom attribute, select it and click **Delete**.

Virtualization Support – VMware ESX (and Solaris 10)

Opware supports VMware ESX 3.0 as an operating system that can host virtual servers. You can provision VMware ESX on a bare metal server to run as an VMware ESX hypervisor, and you can provision any VMware ESX-supported operating system on a VMware ESX guest virtual server, just as you would provision a physical server.

Because the OS provisioning process for VMware ESX follows the same general process as provisioning Linux (with some minor differences), you provision VMware ESX as you would Linux. Any differences in the provisioning process between Linux and VMware ESX are documented separately.



Opware OS Provisioning does not provision HP-UX or AIX operating systems out of the box; however, the Opware SAS can be integrated with Network Installation Management (NIM) to provision AIX and Ignite-UX to provision HP-UX. See *Opware[®] SAS Administration Guide* for more information.

See the *Opware[®] SAS Planning and Installation Guide* for information about how OS provisioning is configured during Opware SAS installation.

Hardware Support in OS Provisioning

This section provides information on hardware support in OS provisioning within Opware SAS and contains the following topics:

- Overview of Hardware Support in OS Provisioning
- PXE Images for Windows and Linux or VMware ESX

- Windows and Linux or VMware ESX Boot Images
- NIC Support in Windows Boot Images
- Adding NIC Support to a Windows Boot Image
- Sample Mapfile
- Sample Mapfile for an Intel 8255x-based PCI Ethernet Adapter
- Prerequisites for Creating Windows Boot Images
- Creating a Windows Boot Image
- Updating PXE Image for Windows
- Adding Hardware Support to a Linux or VMware ESX Build Image
- Creating a Linux or VMware ESX Boot Image
- Example: Usage of the OPSWlinuxbootiso Utility

Overview of Hardware Support in OS Provisioning

OS Provisioning supports a broad range of hardware platforms out of the box, but it also provides an OS Provisioning feature for hardware models not initially supported. To prepare your system for OS Provisioning, you must package and upload system utilities provided by the server manufacturer into Opware SAS. At a minimum, you must update the boot processes for Windows and Linux (Opware Boot Floppies or CDs and the PXE boot system) to support the new hardware. Additionally, you might have to update the Linux build images.

See “Adding NIC Support to a Windows Boot Image” on page 201 and “Adding Hardware Support to a Linux or VMware ESX Build Image” on page 205 for more information.

PXE Images for Windows and Linux or VMware ESX

OS Provisioning supports booting new x86-processor-based servers with the Preboot Execution Environment (PXE) protocol.

When Opware SAS was installed with the Opware Installer, a default boot image was added to the PXE system for Windows and for Linux so that new servers can be booted for the first time over the network. The boot image is used by Opware SAS as the second stage PXE image for PXE network bootstrap programs such as PXELinux.

For Linux, Opware SAS includes a boot image that contains the `bootnet.img` CD for Red Hat Linux AS 3.0. The image has changed to the `syslinux.cfg` and `boot.msg` files; however, the kernel and `initrd.img` are identical to the files on the Linux OS media.

The default boot images include common NIC drivers for many hardware makes and models. Opware SAS uses these NIC drivers to boot new x86-processor-based servers for the first time.

The Linux PXE image and Linux Boot CD contain the same NIC drivers included on the Red Hat Linux AS 3.0 installation CD-ROM or DVD.

Windows and Linux or VMware ESX Boot Images

For environments with servers that do not support network boot technology, Opware SAS supports floppy-based or CD-based booting. This applies to both Linux and VMware ESX operating systems.

You can create a Windows boot floppy from the default boot image for Windows. Opware SAS includes the Opware Build Image Administrator, a tool for creating a boot floppy for Windows.

See “Creating a Linux or VMware ESX Boot Image” on page 206 for more information.

NIC Support in Windows Boot Images

Opware SAS includes a default set of common NIC drivers for many hardware makes and models. If the NIC drivers you need for your environment are not included in the default set, you must add them to the boot image for Windows.

The Opware Build Image Administrator has the ability to dynamically detect your server's PCI network adapter. It does this by scanning the PCI bus for PCI information and comparing the information against each entry in a driver catalog until it finds a match. The driver catalog is constructed each time you create a boot image with the Opware Build Image Administrator.

Each properly formatted cabinet file in the directory `\content\drivers\ndis` under the Opware Build Image Administrator directory is included as an entry in the driver catalog.

Table 3-12: NIC Drivers Included with the Windows Boot Image

DRIVER NAME	DESCRIPTION
B57	Broadcom NetXtreme Gigabit Ethernet NDIS2 Driver v5.20 (021025)
DC21X4	Digital 2104x/2114x 10/100 mbps Ethernet Controller v3.00
E1000	Intel 8254X Based Adapter (pro/1000 gigabit) v1.28 040302
E100B	Intel(R) PRO PCI Driver v4.35 042902
EL59X	3Com DOS NDIS driver for 3C59X Family Adapters v1.2f
EL90X	3Com Etherlink PCI DOS NDIS driver v5.2.2
ELNK3	3Com DOS EtherLink 10 ISA (3C509b) Network Driver v3.1
ELPC3	3Com Megahertz Ethernet PC Card 589E DOS Netw. Driver v1.9.002
ELPC575	3Com Megahertz 10/100 LAN CardBus PC Card DOS NDIS driver v3.4b
FA31X	Netgear FA310TX Fast Ethernet PCI Adapter
FETND	VIA Rhine Family Fast Ethernet Adapter Driver v4.05
N100	Compaq Fast Ethernet and Gigabit NDIS 2 NIC Drivers 7.0a (25Jan02)
NE2000	Microsoft NE2000 NDIS Driver
NETFLX3	Compaq NetFlex-3 DOS NDIS 2.02 driver
PCNTND	AMD PCNet Family Ethernet Adapter NDIS v2.0.1 MAC Driver v3.12
RTSND	Realtek RTL8139/810X Family PCI Fast Ethernet v3.23 07/28/99
SMC9432	SMC EtherPower II 10/100 (9432TX) v1.02c (970605)

If the NIC drivers that you need for your environment are not included in the default set, you must perform the following tasks:

- Add them to the boot image for Windows, Linux, or both.

Update the Windows or Linux boot image in the PXE system with the new boot images.

Adding NIC Support to a Windows Boot Image

Before you perform this procedure, you must obtain the appropriate NDIS2 network drivers and `protocol.ini` file from the manufacturer of the card.

Perform the following steps to add NIC support to a Windows boot image:

- 1** Create a temporary working directory for accumulating files that becomes part of the cabinet file.
- 2** Place NIC drivers and the `protocol.ini` files in the temporary directory.
- 3** Create a text file called `ndis.pci` in the temporary working directory.
- 4** Using a PCI bus scanner, determine the PCI vendor ID and device ID of the NIC card.
For example, the 8255x-based PCI Ethernet Adapter from Intel has vendor ID 8086 and device ID 1229.
- 5** Using the vendor ID and device ID you obtained for the NIC, construct the mapfile `ndis.pci`.

In the mapfile, lines that begin with a semicolon (;) are treated as comments and ignored.

The sample mapfile in this section contains comment lines so that you can use it as a header for your mapfile.

- 6** Create a file named `ndis.txt` in the temporary directory that contains the following single line of text:

```
[basename of cabinet file] "[Driver description string]"
```

The information in this file is used to make up a selection list if the PCI adapter cannot be automatically detected.

Example `ndis.txt` file for the `E100B.CAB`:

```
E100B "Intel(R) PRO PCI Driver v4.35 042902"
```

- 7** Create the cabinet file by using `cabarc` and copy the cabinet file to the directory `.\content\drivers\ndis` under the Opware Build Image Administrator directory. (`Cabarc` is a Microsoft utility that creates, extracts, and lists the contents of cabinet files.)

```
E:\temp\temp_cab>cabarc N e100b.cab *
Microsoft (R) Cabinet Tool - Version 5.2.3718.0
Copyright (c) Microsoft Corporation. All rights reserved.
Creating new cabinet 'e100b.cab' with compression 'MSZIP':
-- adding e100b.dos
-- adding e100b.ini
```

```
-- adding ndis.pci
-- adding ndis.txt
Completed successfully
```

Sample Mapfile

Modify the contents of this sample mapfile. This sample mapfile contains comment lines so that you can use it as a header in the mapfile that you create.

```
; Mapfile for PCISCAN "PCI PnP for DOS"
;
; Syntax:
;   ret="string_to_return"
;   ven=<vendorID> ["Vendor description"]
;   dev=<deviceID> ["Device description"]
;
; Example:
;   ret="aspi8dos.sys"
;   ven= 9004 "Adaptec"
;   dev= 7078 "Adaptec AIC-7870 PCI SCSI Controller"
;       7178 "Adaptec AHA-294X/AIC-78XX PCI SCSI Controller"
;       7278 "SCSI Channel on Adaptec AHA-3940/3940W PCI SCSI
;       Controller"
;       7478 "Adaptec AHA-2944 PCI SCSI Controller"
;       7578 "SCSI Channel on Adaptec AHA-3944 PCI SCSI
;       Controller"
;       7678 "Adaptec AIC-7870 based PCI SCSI Controller"
```

Sample Mapfile for an Intel 8255x-based PCI Ethernet Adapter

```
ret="E100B"
ven=8086 "Intel"
dev=1002 "PRO 100 Mobile Adapters"
  1031 "PRO/100 VE Network Connection"
  1032 "PRO/100 VE Network Connection"
  1035 "PRO/100 VM Network Connection"
  1036 "82562EH based Phonenumber Network Connection"
  1038 "PRO/100 VM Adapter"
  1039 "PRO/100 VE Network Connection"
  103b "PRO/100 VM Network Connection"
  103c "PRO/100 VM Network Connection"
  103d "PRO/100 VE Network Connection"
  103e "PRO/100 VM Network Connection"
  1059 "PRO 100 Mobile Adapters"
  1229 "8255x-based PCI Ethernet Adapter (10/100)"
  2449 "PRO/100 VE Desktop Adapter"
  2459 "82562 based Fast Ethernet Connection"
```

245d "82562 based Fast Ethernet Connection"

Prerequisites for Creating Windows Boot Images

The Opsware Build Image Administrator is used to create a Windows Boot Image that installs the OS Build Agent on servers. The Opsware Build Image Administrator is packaged with MSI.

You must meet the following requirements to use the Opsware Build Image Administrator:

- The machine on which the Opsware Build Image Administrator is installed must have a Python interpreter installed. You can obtain a Python interpreter from ActiveState.
- Opsware SAS must include a default set of common NIC drivers for the hardware makes and models you want to support. If the NIC drivers that you need for your environment are not included in the default set, you must add them to the floppy image.

See “Adding NIC Support to a Windows Boot Image” on page 201 for more information.

Creating a Windows Boot Image

Perform the following steps to create a Windows boot image:

- 1** Download the MSI package that contains the Opsware Build Image Administrator by downloading the file `opswbia-<version>-0.msi` from the SAS Web Client.

Where `<version>` is the latest version of the Opsware Build Image Administrator tool for the release of Opsware SAS installed at your facility. Only one version of the Opsware Build Image Administrator tool is available on the Software Repository.

See “Overview of Package Management” on page 93 in Chapter 2 for information about downloading a package.

- 2** Install the MSI package that contains the Opsware Build Image Administrator tool on a Windows server that has a Python 1.5.2 interpreter.

By default, the Opsware Build Image Administrator is installed in the following directory: `%SystemDrive%\Program Files\OPSWBIA`

- 3** Change directories to the Opsware Build Image Administrator installation directory:

```
\Program Files\OPSWBIA >
```

- 4** Insert a disk into drive A.

- 5** Run the python script `mkimage.pyc`.

```
\Program Files\OPSWBIA > python mkimage.pyc <options>
```

If you do not enter any options, the Opware Build Image Administrator creates a PXE build image file `dosopsw.1` in the current working directory. Enter the `-w` option to write the file `dosopsw.1` to a disk.

Options for the Opware Build Image Administrator

You can use the options described in Table 3-13 when you run the Opware Build Image Administrator from the command line.

Table 3-13: Opware Build Image Administrator Command Line Options

OPTION	DESCRIPTION
<code>-a <drive></code>	Writes the boot image to this drive (Default drive: <code>A</code>).
<code>-c</code>	Makes an El Torito bootable CD image in addition to the boot image.
<code>-d</code>	Enables debugging of the OS Build Agent in the generated image.
<code>-f</code>	Formats the disk first when writing to a disk.
<code>-h <host></code>	Specifies the host name for the Agent Gateway (required option).
<code>-i <file></code>	Specifies the file name for the generated boot image (Default file name: <code>dosopsw.1</code>).
<code>-n <directory></code>	Specifies the directory where the NDIS driver packages are located (Default directory: <code>./content/ndis</code>).
<code>-p <port></code>	Sets the port for the OS Build Agent to use to contact the Agent Gateway (Default port: <code>8017</code>).
<code>-t</code>	Performs a test image generation and does not execute any commands.
<code>-w</code>	Writes the generated image to a disk in the drive specified by the option <code>-a</code> .

Updating PXE Image for Windows

When Opware SAS was installed with the Opware Installer, an image was added to the PXE system by default. You only need to update the PXE image when you have added support for additional NIC drivers to the image.

See “Adding NIC Support to a Windows Boot Image” on page 201.

After adding NIC support to the boot image, install the boot image by using `scp` to copy the image file into the `/opt/opsware/boot/tftpboot` directory on the Opsware Build Server.

Adding Hardware Support to a Linux or VMware ESX Build Image

You can modify Opsware OS Provisioning to add new hardware support to a Linux build image.



Adding hardware support to a VMware ESX build image follows the same general process as for Linux. However, adding drivers to VMware ESX is rarely necessary as drivers for all supported hardware are included in the VMware ESX distribution.

To provision servers with a Linux OS, Opsware SAS uses the two following types of Linux build images:

- **A Linux Boot Image:** Opsware SAS uses a modified version of Red Hat Linux AS 3.0 as a bootstrap image. The Linux Boot Image is loaded on servers when they are booted up for the first time by using the Linux Boot CD or by using PXE. The server appears in the Server Pool list and is ready to be provisioned with an OS.
- **A Linux Build Image that installs the target OS:** Opsware SAS uses this type of Linux Build Image to install the target Linux OS on servers.

To add new hardware support to a Linux Build Image, you must recompile the kernel and modules, and insert the modules into the `initrd.img` file and replace the kernel if it changed.

The Linux Build Images are located on the OS Build Manager host in the following directories:

```
/opt/opsware/buildscripts/linux/bi-<version>
```

Where `<version>` is the version of Linux.

When you modify the Linux Boot Image, include the following options in the kernel:

```
CONFIG_PACKET=y
CONFIG_FILTER=y
```

Setting these options is required if you want to retrieve the Build Manager parameters from DHCP. The existing Linux Boot Image is compiled with these options.

See the Red Hat Linux or SUSE Linux documentation for information about how to add hardware support.

Creating a Linux or VMware ESX Boot Image

Opware SAS includes a command line utility, OPSWlinuxbootiso, that you can use to create a Linux Boot Image on a CD. Running the `mkcdrom.sh` script of OPSWlinuxbootiso creates an ISO file that you can write to a CD.



This procedure describes how to create a boot image for a Linux boot image, but works for VMware ESX boot images as well.

To create a Linux boot image, perform the following steps:

- 1** In the SAS Web Client, search for the package name `OPSWlinuxbootiso*` and the operating system Red Hat Enterprise Linux AS 3.0.
- 2** Download the package to a server or desktop running Linux.
- 3** On the server or desktop where you downloaded the OPSWlinuxbootiso utility, verify that version 1.10-4 of the `mkisofs` utility is installed.
- 4** Change to the following directory:

```
cd /opt/OPSWlinuxbootiso
```
- 5** Run the `mkcdrom.sh` script:

```
./mkcdrom.sh <file-name.iso>
```
- 6** At the prompts, enter the following information:
 - The IP address or host name of the core server running the Agent Gateway (default host name: `buildmgr`)
 - The port on the Agent Gateway that is forwarded [default: 8017]
 - The IP address or host name of the Boot Server (default host name: `buildmgr`)
 - The path to the media for the OS Build Agent (default path: `/opt/OPSWboot/kickstart`)
 - The network interface from which to run Linux Kickstart

Example: Usage of the OPSWlinuxbootiso Utility

```

sin: cd /opt/OPSWlinuxbootiso
sin: ./mkcdrom.sh /tmp/boot.iso
Please enter IP or hostname of Agent-Side Gateway [buildmgr]:
Please enter the port on the Agent-Side Gateway that is
forwarded [8017]:
Please enter IP or hostname of Boot Server[buildmgr]:
Please enter path to bootagent media[/opt/OPSWboot/kickstart]:
Please enter which network interface you would like to kickstart
from
(e.g. eth0) just press enter to choose at runtime:
buildmgr
8017
buildmgr
/opt/OPSWboot/kickstart
*****
* Rewritting isolinux.cfg *
*****
*****
* Building iso... /tmp/fooboot.iso
*****
INFO:   UTF-8 character encoding detected by locale settings.
        Assuming UTF-8 encoded filenames on source filesystem,
        use -input-charset to override.
Size of boot image is 4 sectors -> No emulation
Total translation table size: 2048
Total rockridge attributes bytes: 0
Total directory bytes: 2048
Path table size(bytes): 26
Max brk space used 0
1858 extents written (3 MB)
sin:/opt/OPSWlinuxbootiso>

```


Chapter 4: Code Deployment Setup

IN THIS CHAPTER

This section discusses the following topics:

- Opsware Code Deployment Process
- Code Deployment & Rollback Setup



You must have specific permissions to setup code and content by using the Opsware SAS Web Client. Contact your Opsware administrator to obtain the necessary access rights.

Opsware Code Deployment Process

This section provides information on the code deployment process within Opsware SAS and contains the following topics:

- Deploying Code
- Uploading Code and Content to Staging
- CDR Operations and Directories
- CDR Features
- CDR Permissions
- Accessing CDR



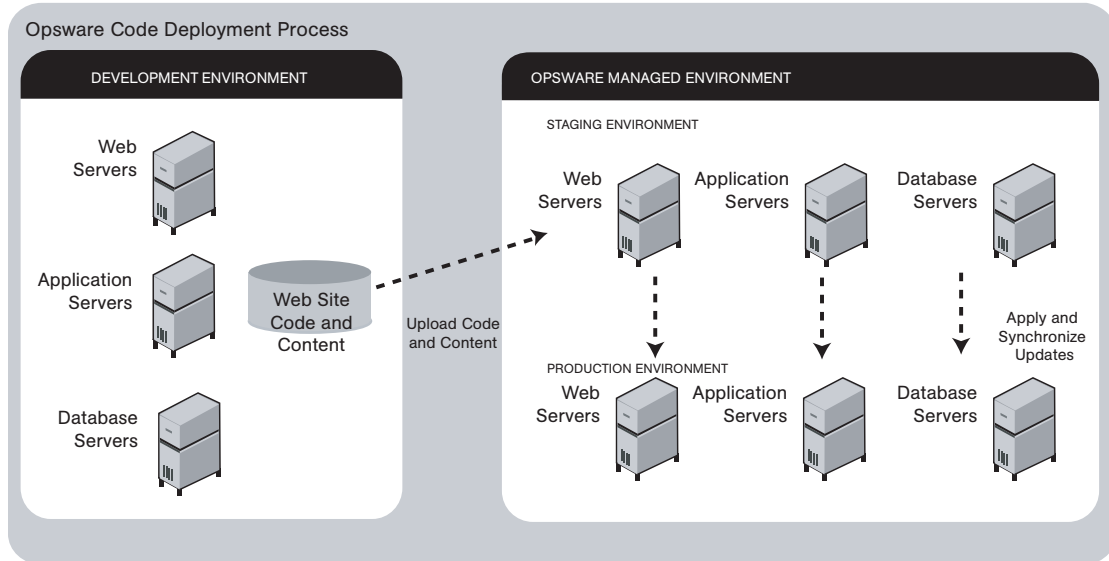
The Code Deployment and Rollback feature is not supported on VMware ESX 3.0.

Deploying Code

The Code Deployment & Rollback (CDR) feature in the SAS Web Client provides tools for deploying new and updated code and content to your operational environment.

The following figure shows the architecture and process for updating a typical server hosted in an Opware managed environment.

Figure 4-1: Typical Code and Content Update in the Opware Managed Environment



The deployment process involves performing the following high-level tasks:

- 1** Determining your application code and content deployment requirements and defining the CDR services, synchronizations, and sequences that you need to support them.
 - Services are defined for each different type of web server or application server applications (for example, WebLogic Server) that is installed on the staging and production hosts in your environment.
 - Synchronizations are defined for each service so that you can update files between the source location and one or more destination production hosts that are running the same service.
 - Sequences are optional but can simplify deployment by grouping a collection of service operations and synchronizations that can be performed as a single task.
- 2** Uploading new or updated code and content to your Opware staging environment.
- 3** After performing any necessary testing, cutting over to the changed code and content on the staging environment.

- 4** As necessary, performing CDR service operations, such as backing up code and content from your live site.
- 5** Performing CDR operations available to synchronize the updated code and content to your production hosts in the Opsware managed environment.
- 6** To simplify subsequent deployments of new code and content, defining sequences that specify a series of service operations and synchronizations you want to perform as a single action.



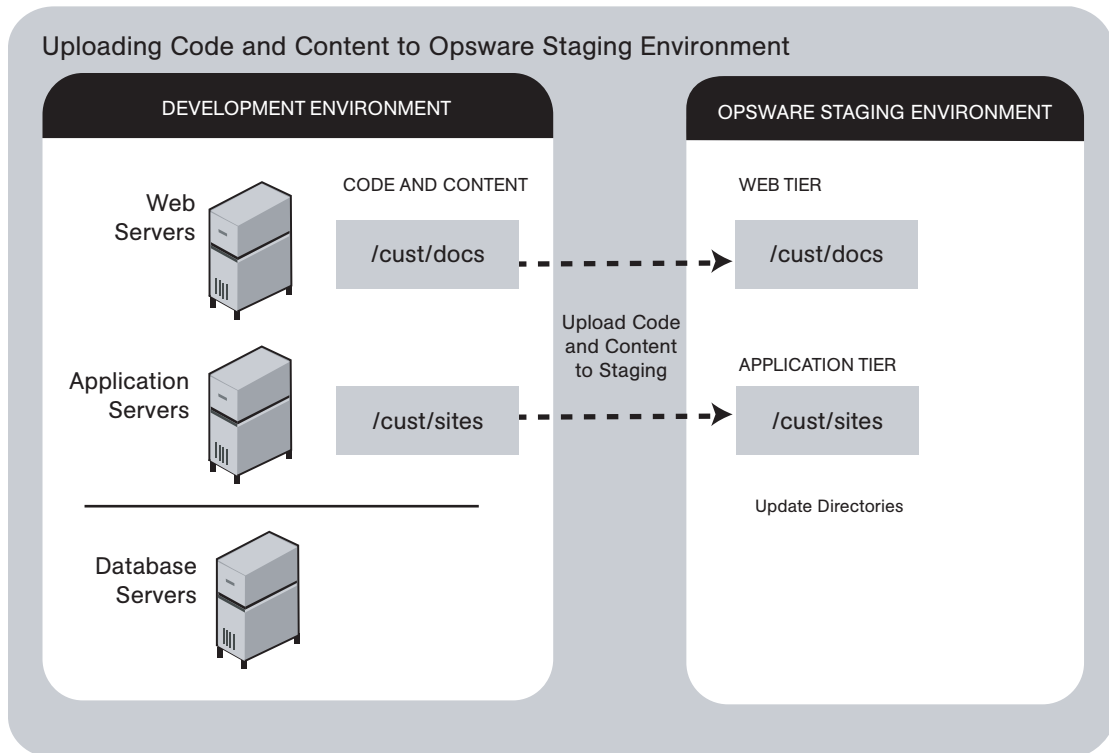
The code and content deployment process that you follow might be different depending on the architecture of your operational environment and your deployment requirements.

Uploading Code and Content to Staging

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

The following figure shows an example of a typical development environment and how your uploaded code and content move to the staging environment.

Figure 4-2: How Code and Content Move to the Staging Environment



After you upload the files and test your changes, you can synchronize updates to the production hosts running your managed environment. You can run specific synchronizations and perform other service deployment operations by selecting CDR menu options available from the SAS Web Client navigation panel.

CDR Operations and Directories

After you upload updated code and content to your Opsware-managed staging environment, you can use the CDR operations to cutover to new code and content, perform host synchronizations, and perform other service operations.

CDR uses the following directories to synchronize and cutover code and content for specified hosts:

- **Live directory:** The directory that stores the actual code and content required to run a live site.

- **Update directory:** The directory written to by CDR synchronizations. Stores only the files that changed between the source host Live directory and the Live directories of the destination hosts.
- **Site Previous directory:** This directory holds all the changes necessary to revert the Live directory back to the state it was in before the last cutover. Like the Update directory, the Site Previous directory only stores the files that changed between the current Live directory contents and its previous state.
- **Site Backup directory:** This directory stores a complete backup of the site. The directory is populated when the user issues a Backup service operation.

When you cutover to new code and content, CDR determines the differences between the new code and content in the current Update directory and the Live directory for your site. The files that are different are synchronized to the Live directory. When you synchronize source and destination hosts, CDR moves modified files from the Live directory on a source host to a directory on a destination host.



You cannot use CDR to automate database pushes. However, you can configure CDR so that you can synchronize modified database script files on different hosts.

CDR Features

CDR offers the following features:

- Provides a single tool for deploying code (such as ASP, JSP, and JAR files) and site content (such as HTML, JPEG, GIF, and PDF files). Using a single tool is helpful when the code and content for your site are intermingled.
- Provides direct control over code and content pushes by making it possible to decide what information to update and determine when and how to perform updates.
- Provides flexibility to accommodate frequent updates to staging and production hosts by enabling more frequent pushes in a shorter period of time.
- Allows verification of file changes between staging and production host directories by creating a manifest of updated files. You can verify changes before cutting over to new code and content.
- Provides administrative service operations, including starting and stopping services, and backing up, restoring, and rolling back code and content to return your site to the previous version.

- Lets you push incremental updates to your site so that only files that have changed are pushed to specified locations on staging or production hosts.
- CDR uses the same authentication and navigation that you use in accessing other information and performing other site operations from the SAS Web Client.

CDR Permissions

As with all other features in Opware SAS, the links that you see on the SAS Web Client Home page and the links that you see in the navigation panel are based on the permissions that you have in combination with the customer you are associated with.

If you do not have permissions for CDR, you cannot see the Code Deployment links on the navigation panel, the link called Deploy Code in the Tasks panel of the SAS Web Client home page appears in italics, and it is not an active link.

If you have CDR permissions to no more than one customer, when you expand the Code Deployment section in the navigation panel, you can see a link called Set Customer. Click that link to view the links to the specific Code Deployment functions that you have permissions for in combination with that single customer.

If you have CDR permissions to more than one customer, you can see a link called Select Customer. Click that link to display a page that shows the customers you are associated with. Select the customer you want to work with. The CDR Home Page appears, with links to the specific Code Deployment functions that you have permissions for. These links are the same functions that you can find in the navigation panel under Code Deployment.



The navigation instructions and screen captures in this chapter show what a user with permissions to all code deployment functions and access to only one customer can see. Consequently, because your permissions and customers might be different, the available menu selections and features that you see might likewise differ.

Accessing CDR

Perform the following steps to access CDR:

- 1** If necessary, click the Code Deployment link in the navigation panel to expand the list of CDR options.
- 2** Click the CDR Home link. The CDR Home Page for [customer name] appears, as the following figure shows.

Figure 4-3: Code Deployment Home Page

CDS Home Page for Main Customer	
LINK	DESCRIPTION
Service Management	Create, Modify, and Delete Service Definitions. Services define the location and commands to manipulate an application on hosts.
Run Service	Perform a service operations on one or more hosts, or request that a service operation be performed on your behalf. Service operations include starting or stopping applications, cutting over or rolling back code, and backing up or restoring code.
Sync Management	Create, Modify, and Delete Synchronization Definitions. Synchronizations define the path for pushing code from a source service host to one or more destination service hosts.
Synchronize	Perform a synchronization to one or more hosts, or request that a synchronization be performed on your behalf.
Sequence Management	Create, Modify, and Delete Sequence Definitions. Sequences allow the grouping of service operations and synchronization operations to define higher level code deployment operations.
Run Sequence	Perform a pre-defined sequence of service operations and/or synchronizations on one or more hosts, or request that a sequence be performed on your behalf.
View History	Get information about previously run Code Deployment Operations.

Depending on your access permissions, the following CDR options appear:

- **Service Management:** Create, modify, or delete service definitions that define the location and commands to manipulate an application on hosts associated with each application instance running in your operational environment.

- **Run Service:** Perform a service operation or request that one be performed.
 - **Sync Management:** Create, modify, or delete synchronization definitions associated with code pushes.
 - **Synchronize:** Perform a synchronization or request that one be performed.
 - **Sequence Management:** Create, modify, or delete sequences of operations.
 - **Run Sequences:** Perform a selected sequence or request that one be performed.
 - **View History:** View information stored in an operations log to determine the status of particular deployment operations, and whether they completed successfully.
- 3** Choose the CDR operations that you want to perform, selecting options from the navigation panel or from the CDR home page.

Code Deployment & Rollback Setup

This section provides information on how to set up and support sites that use CDR for code and content pushes. It contains the following sections:

- Prerequisites for Code Deployment & Rollback
- Code Deployment Configuration Checklist
- Planning and Defining a CDR Configuration
- Configuring a Site to Use CDR for Code and Content Updates
- Code and Content Deployment Requirements
- Overview of CDR Configuration Planning
- Preparing Host Machines
- Creating or Verifying Directories on Hosts
- Initial Content in Directories
- Access Control for CDR
- Defining CDR Services, Synchronizations, and Sequences
- CDR Service Management
- Defining a Service
- Pre-Synchronization and Post-Synchronization Scripts

- Modifying a Service
- Deleting a Service
- CDR Synchronization Management
- Defining a Synchronization
- Modifying a Synchronization
- Deleting a Synchronization
- CDR Sequence Management
- Defining a Sequence
- Modifying a Sequence
- Deleting Sequences
- Verifying and Troubleshooting CDR Configuration

Prerequisites for Code Deployment & Rollback

In configuring CDR for a specific site, you first need to install and configure required software on each of the host machines used in your Opsware managed staging and production environment. Then, you define the set of services, service operations, and staging and production server synchronizations and sequences to make available.

By selecting Service, Synchronization, and Sequence options from the CDR menus, users can either perform operations or request that other authorized users perform them. (Permissions to perform specific CDR operations depend on the code deployment user groups to which individual users are assigned.)

See *Opsware[®] SAS Administration Guide* for information on how to create users and assign the SAS Web Client permissions.



The instructions provided in this section are intended to be platform-neutral. However, platform-specific information and examples are provided where necessary.



The preparation of host machines, directory configuration, and testing should all be carried out during scheduled maintenance windows because modifications made to production machines might cause downtime for the live site.

Code Deployment Configuration Checklist

Before you set up your site to use CDR, collect the following information about the site:

- Names of all host machines used for a site and their designation for use as staging, QA, production, and so forth
- All service instances installed for a site (for example, WebLogic, iPlanet Web Server, and so forth)
- All top-level code and content directories that are used by each of the service instances. (Directories are based on the service or service instance and are the same on all host machines where a particular service or service instance is installed.)
- The name of the machine and directory location where site code and content is uploaded. (This is the host and directory location where you upload files from your own development environment, using an Opware-supported content deployment tool such as OpenDeploy, scp, or rsync over SSH.)



Make sure that you have identified an appropriate process to upload changed code and content from your development environment and check that the appropriate firewall conduits and connections are created to allow uploading changed code and content into the site.

- For a new site deployed in an Opware managed environment, you should also obtain a copy of your site's current code and content to preload into directories prior to using CDR. Preloading code and content shortens the time required to complete updates the first time that you use CDR to perform synchronizations.

Planning and Defining a CDR Configuration

The overall process for planning and defining a CDR configuration and using CDR to define services and synchronizations for your site consists of the following tasks:

- 1** Determine your site's code and content deployment requirements.
- 2** Define the services and synchronizations that are needed to support your site requirements. Optionally, define any sequences of both services and synchronizations that you would like users to define as sequences so users can perform them in a single step.
- 3** Upload new or updated code and content to the Opware staging environment.

- 4** Cutover to the changed code and content on the staging environment and perform any required testing.
- 5** As necessary, you can also set up email notification to send requests to select users to perform CDR service operations, such as backing up code and content from their live site and synchronizing the updated code and content to their production hosts.

Your Opsware administrator determines the responsibilities that different users have pertaining to synchronizations and other service operations performed for a specific site.

Configuring a Site to Use CDR for Code and Content Updates

The following summary shows the steps involved in configuring a site to use CDR for code and content updates, code pushes, and other service/synchronization operations.

The sections that follow described each of the steps in detail:

- 1** Determine your code and content deployment requirements.

Determine the responsibilities that users who are assigned to perform synchronizations, sequences, and other service operations will have.
- 2** Plan your CDR configuration.

Create diagrams of your site's host configuration, specifying synchronization and service descriptions, including any special service operations that you want carried out when a specific synchronization or sequence is performed.

See "Overview of CDR Configuration Planning" on page 221 in this chapter for information about how to document your CDR configuration and the services, synchronizations, and sequences that you are creating for your site.
- 3** Set up access control for CDR.

Have your Opsware administrator create and add users to user groups to create, edit, request, or perform CDR services, synchronizations, and sequences. (User groups that have specific permissions to perform CDR operations are predefined.)
- 4** Create Services and Synchronizations in CDR.

Using the service, synchronization, and sequence documentation defined for your site, create each service, synchronization, and sequence in CDR. Assign the user groups required to access each service, synchronization, or sequence when a user logs into the SAS Web Client

See “Defining CDR Services, Synchronizations, and Sequences” on page 229 in this chapter for more information.

- 5 Verify that the following port is accessible between the server you will push code from and the server where you will push code to:

- `telnet <staging_server> 1002`
- `telnet <production_server> 1002`

- 6 Configure email notification addresses.

Specify the email addresses where notifications are sent when users request that a service operation, synchronization, or sequence be performed on their behalf.

- 7 Test CDR setup and configuration.

After all services, synchronizations, and sequences are defined, and user accounts and permissions are set up in the SAS Web Client, test the operations available for each service, synchronization, and sequence defined in CDR. Uploading both code and content changes from your site development environment, verify that CDR can be used to update services on all staging and production hosts for which synchronizations are defined.

Code and Content Deployment Requirements

Discover your exact deployment requirements, and determine the responsibilities that users who are assigned to performing synchronizations and other service operations will have.

Depending on the setup of your site, you might want certain users to perform routine content updates to your site and assign responsibility for more critical application code changes to other users who will, for example:

- Perform service operations for your production site.
- Synchronize updated code and content to your production site.
- Run sequences that perform a sequence of service operations and sequences as a single step.

CDR lets you send email requests to specific users, notifying them to perform a synchronization, sequence, or other service operation.

The options that are available to users when they access CDR depend on the user groups and permissions the users have been assigned.

See *Opware® SAS Administration Guide* for information on how to create users and assign SAS Web Client permissions.

Overview of CDR Configuration Planning

Before you can use CDR, define the services and synchronizations you need to update and maintain your site. You define individual services based on each specific Web server or application server application (for example, WebLogic Server) that is installed on the staging and production hosts. You define synchronizations so that you can update files for a given service between the source location and one or more destination production hosts.

To define CDR services and synchronizations, you need to know:

- What the code and content directories are for each host
- Which hosts for your site are staging, production, and QA
- What services (for example, Web server or application server programs) are installed on each server

When you log into the SAS Web Client, CDR displays predefined services and synchronization that are available for your site. You see only the services and synchronization that you have authorization to perform because of your user group membership.



The operations that you need to perform are specific to the service (web server or application server instance) for which you are updating code or content and to the particular host.

Before you use CDR to define services and synchronizations, you should document and diagram your site's configuration. That way, when you start defining services and synchronizations, the process is likely to go more smoothly.

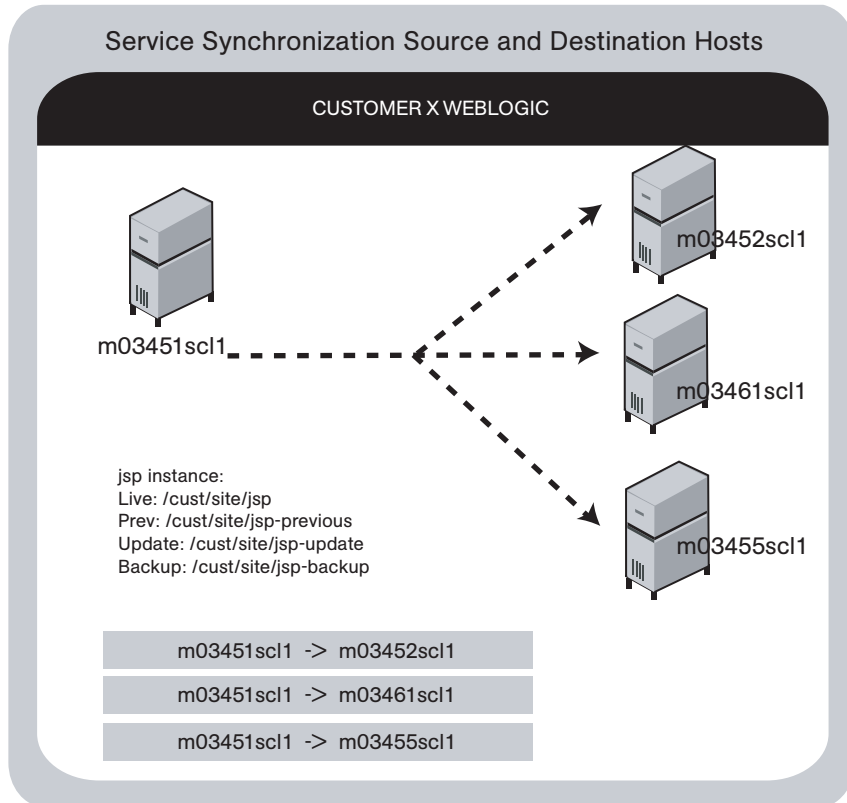
Planning Your CDR Configuration

To plan your CDR configuration, complete the following tasks for every instance of a service (for example, WebLogic application server or iPlanet Web server):

- 1** Create a diagram of your site's host configuration for the service, designating the source and destination host names for any synchronizations that you want to define.

Figure 4-4 shows an example of a typical synchronization diagram defined for a specific service, in this case, a WebLogic (jsp) application instance.

Figure 4-4: Service Synchronization Source and Destination Hosts



In the diagram, all three arrows are part of the same synchronization and specify update paths from the source to the destination host. The diagram also specifies the site's Live, Previous, Update, and Backup directories used by the instance in performing synchronizations, backup, restore, and rollback operations.

- 2** List the service directories, scripts, and any special operations or procedures to be performed for any synchronization of code or content for that service.

Table 4-1 shows the information that you can specify for a service defined in CDR.

Table 4-1: Table of Information to Specify for Code Deployment Service

SECTION OF PAGE	FIELD NAME
Service	
	Name (required)
	Type (required)
Service Commands	
	Start
	Stop
	Pre-cutover
	Post-cutover
	Pre-rollback
	Post-rollback
	Pre-Sync To Update
	Post-Sync To Update
	Pre-Sync To Live
	Post-Sync To Live
	Pre-backup
	Post-backup
	Pre-restore
	Post-restore
Service Directories	
	Live Directory (required)
	Update Directory (required)
	Backup Directory (required)
	Previous Directory (required)

Table 4-1: Table of Information to Specify for Code Deployment Service (continued)

SECTION OF PAGE	FIELD NAME
Service Hosts	
	Hosts
Roles	
	Perform Role Name (required)
	Request Role Name (required)
Service Options	
	CC Operation Requests To

- Determine the name that you want to give the synchronization when you create it by using CDR, for example, WebLogic Sync (Staging to Production). You should designate the user groups whose members can request or perform the synchronization.

Table 4-2 shows information that you can specify for synchronizations defined in CDR.

Table 4-2: Table of Information to Specify for Synchronizations

SECTION OF PAGE	FIELD NAME
	Name (required)
	Associated Service Name
	Source Host Type (required)
	Source Host (required)
	Destination Host(s) Type (required)
	Destination Host(s) (required)
	Perform Role Name (required)
	Request Role Name (required)
Options	
	CC Operation Requests To
	Strict Synchronization

- 4 Repeat the process to create additional diagrams for each instance available in your site environment for which you want to be able to push code or content.



Documenting your plans for code and content deployment for your site will simplify the process of defining new services and synchronizations when you access CDR.

Distributing this information to other users provides useful documentation of your site's configuration, so that everyone involved in code and content deployment for your site understands what services are defined and what synchronizations are available to push code and content.

Preparing Host Machines

After you determine the CDR configuration of services and host machines for your site, perform the following tasks:

- Prepare each host machine in that configuration so that CDR can perform the synchronizations and service operations that you define.
- Create or verify the existence of Live, Previous, Update, and Backup directories on all source and destination hosts.

Creating or Verifying Directories on Hosts

Before you perform synchronization, you must create or verify that the Live Directory, Previous Directory, Update Directory, and Backup Directory already exist on all source and destination hosts. Using the list of host machine names that you collected for your site, log into each of the hosts and check that all the directories already exist or create them.



To determine disk space requirements for a site, you can estimate that CDR requires between two and four times the total size of code and content installed on a particular host, depending on CDR usage factors such as number of changed files between synchronizations and cutovers, use of backup features, and so forth.

- The Live Directory is the directory used for deployed code and content on your site (for example, `/cust/docs` for a Web server, `/cust/site` for an application server like WebLogic).

- The Previous Directory is the directory that records the difference between the current Live Directory's code and content and the version of the site as it existed prior to the last cutover.
- The Update Directory is the directory written to by a CDR synchronization that records the difference between the Live Directories on the source and destination systems.
- The Backup Directory is the directory used to store files when users request a backup copy of the current code and content Live directory.



Ownership of the Live, Previous, Update, and Backup directories is not important because CDR software used to perform service operations and synchronizations runs as root.

Initial Content in Directories

After you create directories on all the hosts designated for a new site deployment (staging, QA, production hosts), you should populate the Live directories on each host with the initial code and content for the site.



Archive a copy of the site files and use a file transfer utility to perform the initial site upload. Using CDR synchronization to initially populate directories has significant overhead and might take longer to perform than directly copying the initial site code and content.

This step (populating directories) only applies to setting up new sites, because current code and content for your site is already available on the Opware staging and production hosts.

In a Unix environment, you can tar the files and use scp to perform the initial site upload into each host Live directory. In a Windows environment, use the Windows file transfer utility to do the initial site upload when you configure host machines for a new deployment.

Access Control for CDR

CDR uses the SAS Web Client authentication to control users' access and ability to perform service operations and synchronizations. Specific permissions to perform code deployment operations are based on a user's membership in predefined CDR user

groups, which an Opsware administrator defines in the Administration section of the navigation panel. Table 4-3, Table 4-4, Table 4-5, and Table 4-6 provide descriptions of permissions that are associated with predefined CDR user groups.

Table 4-3: Special Code Deployment User Groups

CDR USER GROUP	DESCRIPTION
Super-User	Users in this user group can define, request, or perform any code deployment operation on hosts for any customer.
History Viewer	Users in this user group can view a log of operations (service operations, synchronizations, and sequences) that were executed from the Code Deployment feature. Viewing this information can help you determine the completion status of particular deployment operations.

Table 4-4: Service User Groups

CDR USER GROUP	DESCRIPTION
Service Editor	Users can define services and modify or delete service definitions.
Service Performer (Production)	These users directly perform or request performance of service operations on hosts designated for use in production.
Service Performer (Staging)	These users directly perform or request performance of service operations on hosts designated for use in staging.
Service Requester (Production)	These users directly request performance of service operations on hosts designated for use in production.
Service Requester (Staging)	These users request performance of service operations on hosts designated for use in staging.

Table 4-5: Synchronization User Groups

CDR USER GROUP	DESCRIPTION
Synchronization Editor	Users can define a synchronization, modify, or delete the synchronization definition.
Synchronization Performer	These users directly perform or request performance of synchronization actions.
Synchronization Requester	These users request performance of synchronization actions.

Table 4-6: Sequence User Groups

CDR USER GROUP	DESCRIPTION
Sequence Editor	Users can define sequences, and modify or delete sequence definitions.
Sequence Performer (Production)	These users directly perform or request performance of sequences of actions on hosts designated for use in production.
Sequence Performer (Staging)	These users directly perform or request performance of sequences of actions on hosts designated for use in staging.
Sequence Requester (Production)	These users request performance of sequences of actions on hosts designated for use in production.
Sequence Requester (Staging)	These users request performance of sequences of actions on hosts designated for use in staging.



When a user submits CDR requests asking that a service operation or synchronization be performed on the user's behalf, an email notification is sent to the individuals assigned to perform the requested service operation or synchronization. See *Opware® SAS Administration Guide* for information on how to assign users to predefined CDR user groups.

Each user group is created without any users initially added. Your Opsware administrator can add individual users to each CDR user group to control their permissions to request or perform service operations, synchronizations, and sequences.

See *Opsware® SAS Administration Guide* for information on how to add users to CDR user groups.

When a user selects a CDR option, Opsware SAS determines the user's user group memberships and determines what service and synchronization actions the user can perform. Depending on user group membership, the user can either (1) perform or request performance of a service management operation or synchronization operation, or (2) request that the operation be performed by users specified in an email notification list.

Defining CDR Services, Synchronizations, and Sequences

By using the list of services and synchronizations that you have planned for your site, you can use CDR to create the corresponding service and synchronization definitions in the SAS Web Client.

You should follow this process:

- 1** Create all the services required for your site. Each service is defined in terms of the commands such as start or stop that are required for the associated service instance.
- 2** After you define all services, create all synchronizations that you want to make available. Each synchronization references a specific service and specifies the source host from which the service's directories and files are to be synchronized to one or more destination hosts.
- 3** After you define services and synchronizations, define sequences to specify a sequence of specific service and synchronization operations that you want to perform as a unit.

See "Overview of CDR Configuration Planning" on page 221 in this chapter for information about how to define the services and synchronizes that you need to create for a particular site.

CDR Service Management

The CDR Service Management option lets you create new services or modify or delete existing services. For example, if you have a single instance of a service, you can use CDR to define a single CDR service. If you have five instances of a service, you can define five individual CDR services.

You should also create different services to provide control over services performed on staging hosts versus production hosts. For example, you could define a service that only names staging hosts and specify a perform user group for users who can perform operations for those hosts. You could then define a second service that names all hosts (both staging and production) and limit the perform user group to selected users.

In CDR, every service is defined down to the level of a single command that needs to run during service operations, such as start, stop, pre-cutover, post-cutover, and so forth. Both services and service instances are defined the same way because conceptually there is no difference between them. For example, you can define an Apache service, or several instances of ATG Dynamo or BEA WebLogic in terms of the scripts required to start or stop the service and scripts to perform at pre-cutover, post-cutover, and so forth.

Defining a Service



If you are invoking Python in a CDR service command, you must invoke Python by using a fully qualified path to `python.exe` in the command. If you are migrating to the current version of Opsware SAS from a previous version, you must update any currently defined CDR service commands.

Perform the following steps to define a service:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Service Management option.
- 3** Click the Define a New Service link. The CDR Service Name and Type page appears, as Figure 4-5 shows.

Figure 4-5: CDR Service Name and Type

Service	
Name	<input type="text"/>
Type	<input type="text" value="Select a service type"/>

- 4** Specify the name of the service by choosing a name that users can identify with the corresponding application instance, for example, WebLogic (EJB Instance).

- 5** Specify the type of service that you want to create by selecting the service type from the drop-down list, as Figure 4-6 shows. The drop-down list includes the names of all application instances defined in the Model Repository.

Figure 4-6: CDR Service Commands

Service Commands	
Start	<input type="text"/>
Stop	<input type="text"/>
Pre-cutover	<input type="text"/>
Post-cutover	<input type="text"/>
Pre-rollback	<input type="text"/>
Post-rollback	<input type="text"/>
Pre-Sync To Update	<input type="text"/>
Post-Sync To Update	<input type="text"/>
Pre-Sync To Live	<input type="text"/>
Post-Sync To Live	<input type="text"/>
Pre-backup	<input type="text"/>
Post-backup	<input type="text"/>
Pre-restore	<input type="text"/>
Post-restore	<input type="text"/>

- 6** In the Service Commands section, enter any commands to perform for the specific service or service instance. In each case, you can enter a single command (specifying a fully qualified path) that is run to effect the operation. The same commands and scripts are applied for all hosts where the service is installed.

- **Start and Stop fields:** Specify single commands or scripts that are executed when users choose the Service Management option to start and stop a specified service.
 - **Pre-cutover and Post-cutover fields:** Specify single commands or scripts that are executed before and after a user chooses the Run Service option to cutover code and content changes to Live.
 - **Pre-Rollback and Post-Rollback fields:** Specify single commands or scripts that are executed before and after a user chooses the Service Management option to restore code and content in a service's Live directory from the service's Rollback directory on specified hosts.
 - **Pre-Sync to Update and Post-Sync to Update fields:** Specify single commands or scripts that are executed before and after a user chooses the Synchronize option to synchronize code and content changes to the Update directory on specified hosts.
 - **Pre-Sync to Live and Post-Sync to Live fields:** Specify single commands or scripts that are executed before and after a user chooses the Synchronize option to synchronize code and content changes to the Live directory on specified hosts.
 - **Pre-Backup and Post-Backup fields:** Specify single commands or scripts that are executed before and after a user chooses the Service Management option to back up code and content from a service's Live directory to a Backup directory on specified hosts.
 - **Pre-Restore and Post-Restore fields:** Specify single commands or scripts that are executed before and after a user chooses the Service Management option to restore code and content from a service's Backup directory to the Live directory on specified hosts.
- 7** In the Service Directories section (see Figure 4-7), specify the disk locations for Live, Update, Previous, and Backup directories used by the service (common for all hosts where a given service is installed).
- **Live Directory:** The directory that stores the actual code or content required by a specific service to run a live site.
 - **Update Directory:** The directory written to by CDR synchronizations. Stores files that changed between the source host Live directory and the Live directories on destination hosts where the service is installed.

- **Backup Directory:** The directory written to by CDR backup operations; used by the Restore option to return a service's Live directories to the code and content of a previous backed up version.
- **Previous Directory:** The directory written to by CDR cutover operations; used by the Rollback option to return a service's Live directories to the code and content that existed prior to the last performed synchronization.

Figure 4-7: CDR Service Directories

Service Directories	
Live Directory	<input type="text"/> (Enter full path e.g. /cust/site)
Update Directory	<input type="text"/>
Backup Directory	<input type="text"/>
Previous Directory	<input type="text"/>

- 8** In the Service Hosts section, select all hosts on which this service is running. You can use the Shift and Control keys to select multiple hosts. See Figure 4-8.

These servers have a use field that has Code Deployment selected in Server Attributes.

The servers also have a state of OK. If you changed the use of a server by using the SAS Web Client, click **Refresh** to update the host list.

Figure 4-8: CDR Service Hosts

Service Host(s)	
Hosts	<input type="text"/> m0178whitesox.cust.custqa4.com m072.goldsox.qa.opsware.com

- 9 In the Roles section, specify the CDR user groups whose members you want to perform or request operations for the specific service. The Perform Role name determines the user group whose members can perform or request that select staff, or your Operations Center, perform a specific operation associated with the service. The Request Role Name specifies user groups whose members can request only an operation, such as start or stop for a service. See Figure 4-9.

See "Access Control for CDR" on page 226 in this chapter for information about the description of CDR user groups that you can specify for the Perform Role and Request Role names.

Figure 4-9: CDR Roles for Performers and Requesters

Roles	
Perform Role Name	Select a role
Request Role Name	Select a role

Figure 4-10: Email Addresses to Copy CDR Operation Requests to

Service Options	
CC Operation Requests To	(xxx@xxx.com,yyy@xxx.com ...)

- 10 In the Service Options section (see Figure 4-10), specify any email address contacts that you want to notify for any service operation requests.

Specifying email notifications allows flexibility in assigning requests to select members of your staff or your Operations Center.

- 11 When you finish making entries to define a new service, click **Save**.

CDR verifies that the service name you specified is unique and then saves the new service definition data in the Model Repository.



To save defined services, you must select at least one host name and provide entries for the Service Name, Service Type, Start Service, Stop Service, Perform Role, and Request Role fields.

Pre-Synchronization and Post-Synchronization Scripts

Pre- and post-synchronization scripts only run on destination hosts.

On Windows machines, you can use the post-cutover command, for example, to specify a command that performs Windows object registration (among other tasks). In that case, you might define a post-cutover script that (1) lists all files in a directory and (2) passes all files with the .dll extension to `regsvr32.exe` and passes all files with the .msi extension to `msiexec.exe`. Performing these steps registers and un-registers COM objects. A similar script can be developed to de-register and register COM+ objects. This script can be named in CDR and must then be placed on all hosts on which the service could run.

You can specify the instance name as a command line argument in Start and Stop commands or scripts for services that describe instances of the same service running on the same hosts. (You need to create different services for each service instance running on the same hosts because the directories and start and stop script calls used by each instance are different.)

Start and Stop and other service command or script entries: If you need to perform operations that require more than a single command, you should define a sequence of commands in a single script file and then specify that script in the CDR service definition.

Modifying a Service

Occasionally, you need to modify an existing service, for example, to change assigned hosts, make updates to scripts, or make other changes to the attributes of the service.

Perform the following steps to modify a service:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Service Management option.
- 3** Click the Modify an Existing Service link.
- 4** Select the name of the service that you want to modify.
- 5** Update the field entries that you want to modify, and then click **OK**. A confirmation page appears.

You can modify all field entries that define a service except for the Service Type field. If you modify the Service Name field to rename a service, CDR confirms that the new name is not already in use.

CDR deletes synchronizations associated with a service when the following modifications are made:

- When a user removes a host name from the list of hosts defined for a service and that host name is a source for a synchronization, that synchronization is also removed when the service definition is saved. If that synchronization is used by a sequence, then that sequence is also removed.
- When a user removes a host name from the list of hosts defined for a service, and that host name is the last remaining destination for a synchronization, that synchronization is also deleted when the service definition is saved. If that synchronization is used by a sequence, then that sequence is also removed.
- When a user removes a host name from the list of hosts defined for a service, and that host name is the last host in a sequence step, then the whole sequence is deleted when the service definition is saved.

Deleting a Service

CDR allows you to delete services and remove their stored definition from the Model Repository.

Perform the following steps to delete a service:

- 1** If necessary, click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Service Management option.
- 3** Select the Delete a Service option.
- 4** Select the check boxes next to the services that you want to delete and click **Delete**.

CDR prompts you to confirm the deletion.

- 5** Click **OK**. CDR removes the services that you chose to delete.

If you request deleting a service definition, CDR displays a confirmation box that indicates that any associated synchronizations or sequences are also deleted when it deletes the service.

CDR Synchronization Management

The CDR Sync Management option lets you create, modify, or delete synchronizations so that you can update files for a given service between a source host location and one or more destination hosts. For example, in setting up a synchronization for a WebLogic application server instance, you can create a synchronization to transfer updated files between a staging host and the production host machines used to run your site.

When defining a synchronization, you first select the service, then specify the source and destination hosts you want to synchronize. In addition, you specify the CDR user groups that can perform or request the synchronization. You can also specify options such as addresses for email notification of synchronization requests and how synchronizations are performed (Strict Synchronization transfers updated files and removes deleted files from destination hosts.)

Defining a Synchronization

Perform the following steps to define a synchronization:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Sync Management option.
- 3** Select the Define a New Synchronization option.
- 4** Select the service to which you want to add a synchronization.

CDR displays a page on which you can define a new synchronization, choose source and destination hosts, and specify other synchronization options. See Figure 4-11.

Figure 4-11: Define a New Synchronization Page

		Update
Name	<input type="text"/>	
Associated Service Name	WebLogic CustApp (Application)	
Source Host Type	Please select ▼	
Source Host	Please select a host type ▼	
Destination Host(s) Type	Please select ▼	
Destination Host(s)	Please select a host type <input type="text"/>	
Perform Role Name	Select a Role ▼	
Request Role Name	Select a Role ▼	
Options		
CC Operation Requests To	<input type="text"/> (xxx@xxx.com, yyy@xxx.com ...)	
Strict Synchronization	<input type="checkbox"/>	
		Save Cancel

- 5** Specify the name of the synchronization, choosing a name by which users can identify the type of synchronization being performed, for example, WebLogic Sync (Staging to Production).
- 6** Specify the Source and Destination Host Types, choosing the type from the drop-down lists, which display all values stored in the Model Repository. These values are editable using Server Attributes.



You need to specify a Host Type before any hosts are displayed in the Source or Destination Host lists.

- 7** Specify the single Source Host for the synchronization from the list of hosts stored in the Model Repository that match the value that the Source Host Type specified.
- 8** Specify one or more Destination Hosts for the synchronization from the list of hosts stored in the Model Repository that match the value that the Destination Host Type specified.



Use the Shift and Control keys to select multiple destination host machines.

- 9** In the Perform Role Name and Request Role Name fields, select the CDR user groups that you want to allow to perform or request operations for the synchronization. The Perform Role Name determines the user group whose members can perform, or request that another member perform a particular synchronization. The Request Role Name specifies user groups whose members can request that authorized individuals perform synchronizations.

See “Access Control for CDR” on page 226 in this chapter for information about a description of CDR user groups that you can specify for the Perform Role and Request Role Names.

- 10** In the Synchronization Options section, specify any email address contacts that you want notified of any synchronization requests.
- 11** Select the Strict Synchronization check box to specify that files deleted from the source host are also removed from corresponding directories on destination hosts defined in the synchronization. (Otherwise, if unchecked, the synchronization affects only files that are new or have changed between the source host and destination hosts and files removed from a source host are not removed from destination hosts.)
- 12** When you finish making entries to define a new synchronization, click **Save**. CDR verifies that the synchronization name that you specified is unique and then saves the new synchronization definition data in the Model Repository.



To save a new synchronization, you must specify a unique synchronization name, the source host and at least one destination host, and user groups that can perform or request synchronizations.

Modifying a Synchronization

Occasionally, you need to modify an existing synchronization, for example, to change source or destination hosts or make other changes to attributes of the synchronization.

Perform the following steps to modify a synchronization:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Sync Management option.
- 3** Select the Modify an Existing Synchronization option.
- 4** Select the name of the synchronization you want to modify.
- 5** Update the field entries that you want to modify, then click **OK**. A confirmation page appears.

When you modify a synchronization by removing a host from the list of destination hosts, and that host is the last host in a synchronization sequence step, then that sequence is removed.

You can modify all field entries that define a synchronization except for the Source and Destination Host Type fields. If you modify the Synchronization Name field to rename a synchronization, CDR confirms that the new name is not already in use.

Deleting a Synchronization

CDR allows you to delete synchronizations and remove their definition from the Model Repository.

Perform the following steps to delete a synchronization:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Sync Management option.
- 3** Select the Delete a Synchronization option.

- 4 Select the check boxes next to the synchronization that you want to delete and click **Delete**.

CDR prompts you to confirm the deletion.

- 5 Click **OK**. CDR removes the synchronizations that you chose to delete.



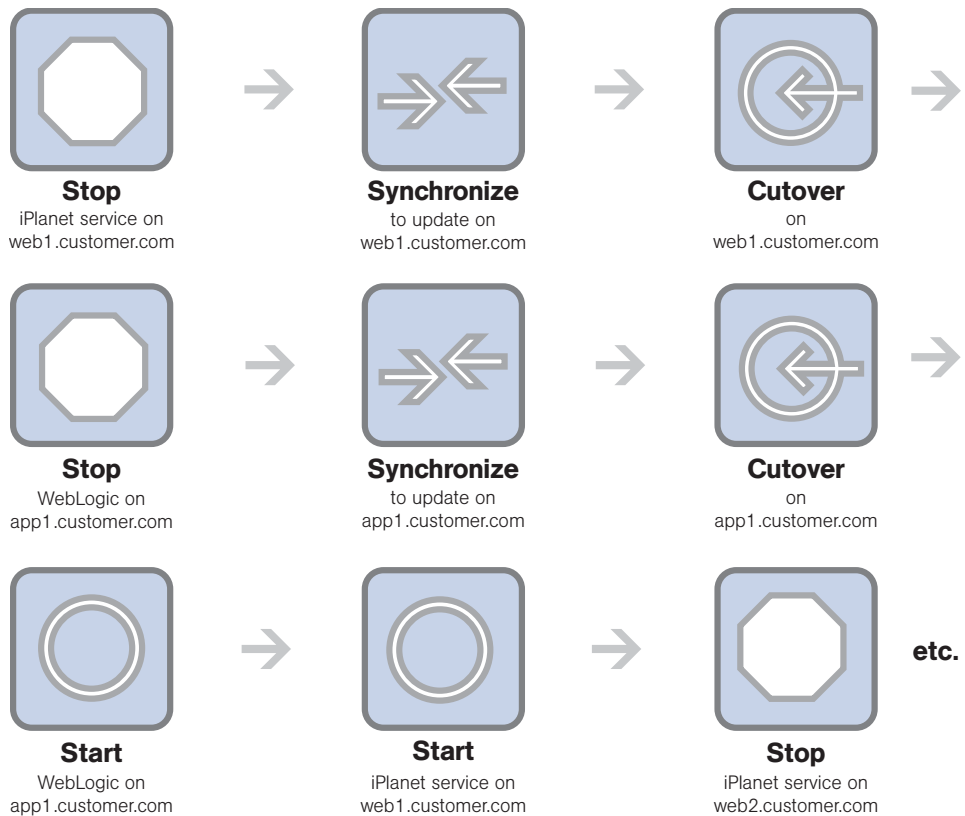
Deleting synchronizations that are used by a sequence causes that sequence to be deleted.

CDR Sequence Management

The CDR Sequence Management option lets you create, modify, or delete sequences of service operations and synchronizations so that you can define meta-operations for CDR.

For example, you can define a sequence to push code from staging to production hosts, stop, cutover, and start a service. A sequence is defined in two parts: the properties of the sequence itself (name, user groups, and so forth) and the steps of the sequence.

Figure 4-12: Example Deployment Sequence



Defining a Sequence

Perform the following steps to define a sequence:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Sequence Management option.
- 3** Select the Create a New Sequence option.

CDR displays a page on which you specify the name of a new sequence, the user groups for the performer and requester for this sequence and email information. See Figure 4-13.

Figure 4-13: Define Sequence Page

Sequence	
Name	<input type="text"/>
Roles	
Perform Role Name	<input type="text" value="Select a role"/>
Request Role Name	<input type="text" value="Select a role"/>
Sequence Options	
CC Operation Requests To	<input type="text"/> (xxx@xxx.com,yyy@xxx.com ...)
Email When Sequence Completes	<input type="text"/> (xxx@xxx.com,yyy@xxx.com ...)
<input type="button" value="Continue"/> <input type="button" value="Cancel"/>	

- 4** Specify the name of the sequence, choosing a name that users can identify with the corresponding operation that this sequence will perform, for example, Push Code to Production.
- 5** In the Roles section, specify the CDR user groups that you want to allow to perform or request execution of the specific sequence. The Perform Role name determines the user group whose members can perform or request that select members of your staff, or your Operations Center, perform this sequence. The Request Role Name specifies user groups whose members can request a sequence.

See “Access Control for CDR” on page 226 in this chapter for information about a description of CDR user groups that you can specify for the Perform Role and Request Role Names.

- 6** In the Sequence Options section, specify any email addresses to whom you want to send sequence operation requests.
- 7** You can also specify email addresses to which notifications can be sent when the sequence is performed and completed. The email contains the status of each step of the sequence that was performed and an indication if it ran successfully.

8 Click **Continue** to save the sequence properties.



To save defined sequences, you must provide entries for the Sequence Name, Perform Role, and Request Role fields.

9 A small window popup on your screen. Use this window to select the operations to add to the sequence. First select the name of the service that you want to operate on in the Service Name drop down menu. See Figure 4-14.



If you use a pop-up blocker, this window will not pop up. You can access it by clicking the hyperlinked word popup in the sentence that says, "Add new operations with the popup window."

Figure 4-14: Sequence Operation Selection Window

Choose an Operation for the Sequence	
Service	Select a service ▼
Synchronization	Select a service ▼
Operation	Select a service ▼
Hosts	<input type="text"/>

Update

Add

10 To add services or synchronizations to a sequence, perform the following steps:

- In both cases, first select a service from the Service drop-down menu, then select a service from the Synchronization drop-down menu.
- To add a synchronization operation, select Synchronize to Update or Synchronize To Live from the Operation drop-down menu, then select one or more destination hosts

for the synchronization from the Hosts select box. Finally, click **Add**. The information about the newly added step appears in the main window.

- To add a service operation, select None from the Synchronization drop-down menu. Then, select the name of the service operation that you want to add from the Operation drop-down menu, and select the hosts that you want to perform the service operation on in the Hosts select box. Finally, click **Add**. The information about the newly added step appears in the main window.

11 Click **Save** to save the sequence.

Modifying a Sequence

Occasionally, you need to modify an existing sequence, for example, to change assigned hosts in a step, add a step, or make other changes to attributes of the sequences.

Perform the following steps to modify a sequence:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Sequence Management option.
- 3** Select the Modify an Existing Sequence option.
- 4** Select the hyperlinked name of the sequence that you want to modify.
- 5** Update the field entries that you want to modify and then click **Continue**.
- 6** Edit any of the sequence steps that you want.
- 7** Click **Save** to save the changes.

Deleting Sequences

CDR also allows you to delete sequences and remove their stored definition from the Model Repository.

Perform the following steps to delete a sequence:

- 1** Click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 2** Select the Sequence Management option.
- 3** Select the Delete a Sequence option.

- 4 Select the check boxes next to the sequences that you want to delete and click **Delete**.

CDR prompts you to confirm the deletion.

- 5 Click **OK**. CDR removes the sequences that you chose to delete.



Deleting sequences has no impact on defined services or synchronizations.

Verifying and Troubleshooting CDR Configuration

After you set up all the CDR services and synchronizations required for your site, and perform all other setup required on host machines in either your development environment or the Opware managed environment, verify operation of the complete configuration.

To verify your CDR configuration, perform the following steps:

- 1 Log into the SAS Web Client with permissions to perform service operations and synchronizations.
- 2 If necessary, click the Code Deployment link in the navigation panel to expand the list of CDR options and select a customer, if necessary.
- 3 Modify files in your staging host's Update (source) directory to enable testing synchronizations.
- 4 Perform all defined synchronizations. After completing the synchronizations, verify that the files, which were modified on your staging source host, were modified correctly in the directories of each destination host.
- 5 Perform all service operations for each defined service to verify the operations of scripts for starting, stopping, cutting over, backing up, restoring, and rolling back updates. Also, verify that all pre-operations and post-operations were successful.
- 6 Verify any sequences that you defined, executing each sequence and then checking that the operations complete successfully.

Chapter 5: Visual Packager

IN THIS CHAPTER

This section discusses the following topics:

- Overview of Visual Packager
- Packaging Server Setup
- Overview of Packages
- Packaging Process
- Ways to Create a Package
- Creating a Package
- Adding New Package Content
- Specifying Options for New Package Content
- Viewing Package Details

Overview of Visual Packager

Visual Packager is an Opsware feature that helps you to create software policies for managed servers. It guides you through the process of creating installable software packages using server or audit and remediation information, such as snapshot results and audit results. File system objects recorded in a snapshot and audits help you define the content of packages, and packages, in turn, can be used to update servers with new server objects.

Server objects can be selectively packaged according to the operating system of the servers that the package will be distributed to. Visual Packager supports Unix and Windows Operating Systems by allowing packages to contain the following objects:

- A Unix package can contain files (including attributes), directories, packages, patches, and patch clusters.
- A Windows package can contain files (including attributes), directories, packages, patches, Windows registry, and Windows services.

Packages can consist of other packages, patches, and non-package content such as the Windows registry, Windows services, and file system objects. When you create a package that contains any of these items, Visual Packager analyzes the objects that you have selected and characterizes them in the following ways:

- When a package contains a package or a patch that does not exist in the Software Repository, Visual Packager lets you provide the missing information for selected packages and patches.
- When a package contains a package or a patch that already exists in the Software Repository and you want to overwrite them, Visual Packager lets you select the source location of the preferred version.
- When a package contains non-package content or patch data, Visual Packager creates a new package that contains them. You can specify a set of options that apply only to the non-package content, such as reboot requirements and pre/post install scripts. This non-package content is considered to be a package that is part of the same software policy that the other packages and patches are attached to.

To get started using Visual Packager, select **Create Package** under the **Actions** menu in the SAS Client.



You must have RPM installed on your packaging server to enable the Visual Packager feature to create an RPM package for Solaris and AIX. The Visual Packager feature does not verify whether RPM is available on the packaging server.

Packaging Server Setup

Visual Packager requires a packaging server for each type of operating system for the packages you plan to create. For example, for Solaris packages you need a Solaris packaging server, and for MSI (Microsoft® Installer Utility) packages you need a Windows packaging server.

When you are using Visual Packager to create a package for a Red Hat Linux operating system, the operating system version and architecture of the packaging server must be identical to the operating system version and architecture that you want the package created for and installed on. Table 5-1 illustrates these requirements.

Table 5-1: Red Hat Linux Packaging Servers and Packages

OPERATING SYSTEM VERSION AND ARCHITECTURE OF THE PACKAGING SERVER	OPERATING SYSTEM VERSION AND ARCHITECTURE OF THE PACKAGE
Red Hat Enterprise Linux 3 AS 32 bit x86	Red Hat Enterprise Linux 3 AS 32 bit x86
Red Hat Enterprise Linux 3 AS 64 bit x86	Red Hat Enterprise Linux 3 AS 64 bit x86
Red Hat Enterprise Linux 3 ES 32 bit x86	Red Hat Enterprise Linux 3 ES 32 bit x86
Red Hat Enterprise Linux 3 ES 64 bit x86	Red Hat Enterprise Linux 3 ES 64 bit x86
Red Hat Enterprise Linux 3 WS 32 bit x86	Red Hat Enterprise Linux 3 WS 32 bit x86
Red Hat Enterprise Linux 3 WS 64 bit x86	Red Hat Enterprise Linux 3 WS 64 bit x86
Red Hat Enterprise Linux 4 AS 32 bit x86	Red Hat Enterprise Linux 4 AS 32 bit x86
Red Hat Enterprise Linux 4 AS 64 bit x86	Red Hat Enterprise Linux 4 AS 64 bit x86
Red Hat Enterprise Linux 4 ES 32 bit x86	Red Hat Enterprise Linux 4 ES 32 bit x86
Red Hat Enterprise Linux 4 ES 64 bit x86	Red Hat Enterprise Linux 4 ES 64 bit x86
Red Hat Enterprise Linux 4 WS 32 bit x86	Red Hat Enterprise Linux 4 WS 32 bit x86
Red Hat Enterprise Linux 4 WS 64 bit x86	Red Hat Enterprise Linux 4 WS 64 bit x86

To use the Visual Packager to create a package for a Red Hat Linux operating system, you must manually install the `rpm-build-<version>` on the packaging server.

The following installation and configuration tasks are required to set up a packaging server:

- You must first install the ISM Development Kit (IDK) on a packaging server by using the SAS Client. The packaging server must already be a managed server.
- You can then configure preferences for a packaging server by using the SAS Client itself.



Opware SAS 6.1, Visual Packager only supports packaging servers that have ISM Tool Version 3.1.x installed. When you upgrade from an earlier version of Opware SAS to Opware SAS 6.1, packaging servers which have ISM Tool 2.0.12 installed are not available in the SAS Client.



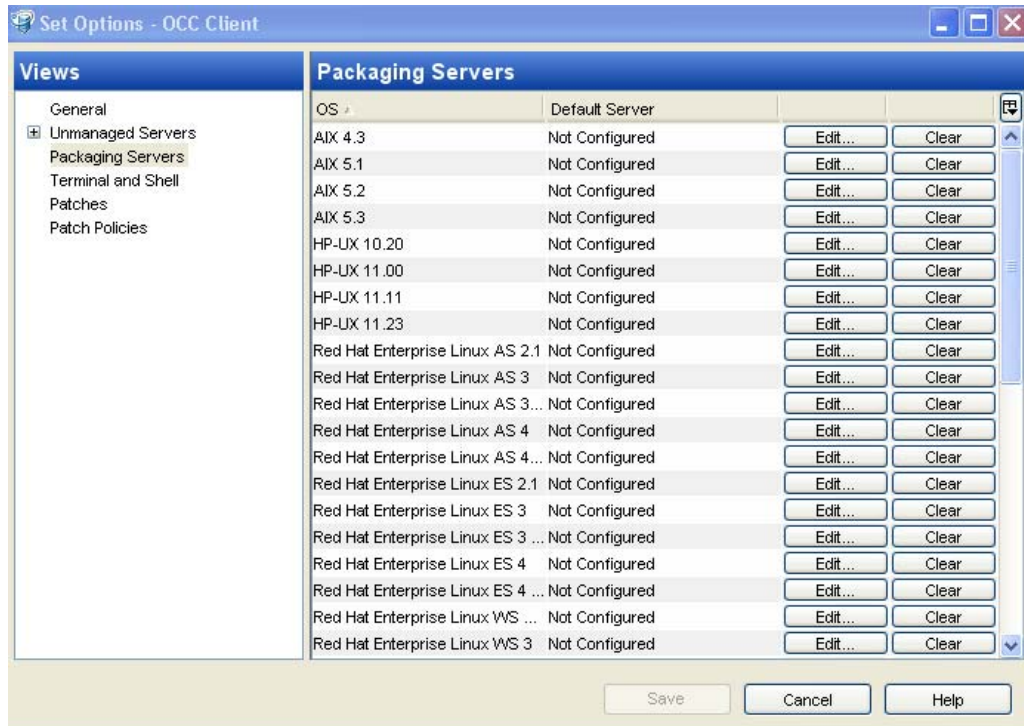
The IDK must be installed on the packaging server in the SAS Client using the ISM Software Policy. Installing the IDK also enables the Visual Packager feature. See the *Opware® SAS Content Utilities Guide* for more information about installing IDK.

Configuring Options for a Packaging Server

To configure a packaging server, perform the following tasks:

- 1 Launch the SAS Client.
- 2 From the **Tools** menu, select **Options**.
- 3 In the Set Options window, select Packaging Servers in the object tree to display a list of managed servers that already have the IDK installed on them. Not Configured in the Default Server column indicates that the server has not been set up as a packaging server for a specific operating system

Figure 5-1: Sample Set Options Window.



- 4 In the content area, select an operating system and click **Edit** to display a list of managed servers for that operating system, or click **Clear** if you want to remove your edits.
- 5 In the Select Server window, select the managed server that you want to configure as the packaging server and then click **Select**.

In the Set Options window, the Default Server column will now display the IP address of the server. Before you configured this as the packaging server, the Default Server column displays Not Configured.



Use the search tool to dynamically filter by entering a server name, IP address, or operating system.

- 6** In the Set Options window, click **Save** to save your changes or click **Cancel** to close this window without saving your changes.

Overview of Packages

A package is installable software that includes server objects that contain applications, data, documentation, and configuration information for a managed server. These server objects can be files, directories, other packages and patches, Windows Registry, Windows Services, and so on. All packages are stored in the Software Repository.

Table 5-2 identifies the types of server objects you can include in a package, according to the operating system you plan to distribute the package to.

Table 5-2: Objects That Can Be Packaged

OBJECT TYPE	UNIX	WINDOWS
Files/Directories	Yes	Yes
Packages	Yes	Yes
Patches*	Yes	Yes
Windows Registry**	No	Yes
Windows Services	No	Yes
MTS/COM+	No	Yes
IIS Metabase	No	Yes



*Patches do not apply to Linux operating systems.

** You can package selected Windows registry keys, such as HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_LOCAL_MACHINE, and HKEY_USERS.

Different types of server objects can be included in a package. For example, you can include multiple files, multiple patches, and multiple Windows services in one package. The IDK creates an installable package of these objects, in the native format of the operating system, such as Solaris Package for Unix, MSI for Windows, LPP for AIX, Depot for HPUX, and RPM for Linux.

In addition to selecting server objects that you want included in a package, you can also select packages and patches that already exist in Opware SAS.

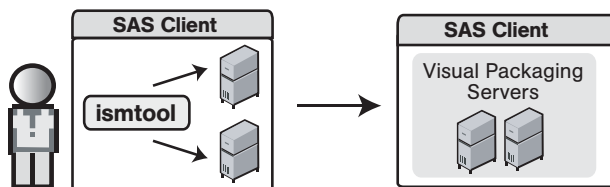
Packaging Process

The packaging process consists of several steps. The packaging process collects your package content, creates a background snapshot of the content, creates an executable wrapper for it, and then attaches it to a software policy in the Library. Figure 5-2 illustrates this workflow.

Figure 5-2: The Packaging Process

VISUAL PACKAGER PROCESS

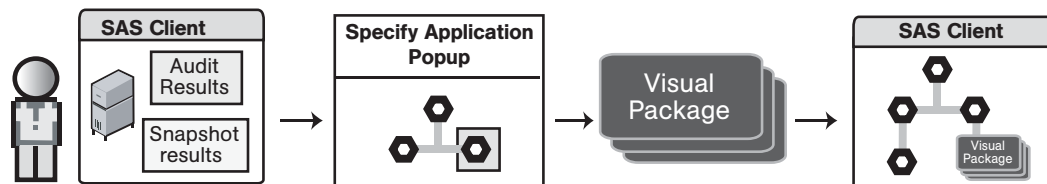
Part A: Set Up Visual Packaging Servers



STEP 1

Adminstrator or User downloads and installs the ismtool to prepare visual packaging servers.

Part B: Create and Deploy Visual Package



STEP 1

User selects a visual package source:
 - Audit Result
 - Snapshot results
 - Server

STEP 2

User selects a location in the Library for the package.

STEP 3

User selects package contents and creates visual package.

STEP 4

The new package is attached to the software policy

During the packaging process, a snapshot of your selected package content is transferred from the source server to the packaging server. This snapshot is used to create an Intelligent Software Module (ISM). The IDK (which must be installed on the packaging server) creates an executable wrapper for your package content. See the *Opware® SAS Content Utilities Guide* for more information.

When you create a package, a software policy is created in a user-defined, package location in the folder hierarchy. Your package is then attached to this software policy.

Ways to Create a Package

In Visual Packager, you can create a package in several different ways, depending on the content source. The content of a package is based on any of the following sources:

- An Opware-managed server
- Audit results
- A server snapshot

Depending on the content source you select, you can launch the Visual Packager process from different windows in the SAS Client.

Creating a Package from a Managed Server

You can create a package from a managed server by using the following different windows in the SAS Client.

In the All Managed Servers window:

- 1** From the Navigation pane, select Devices and then select All Managed Servers.
- 2** From the Content pane, select one or more managed servers.
- 3** Select **Actions** ► **Create Package**.

In the Device Groups window:

- 1** From the Navigation pane, select Devices and then select Device Groups.
- 2** From the Content pane, select one or more servers in a device group.
- 3** Select **Actions** ► **Create Package**.



You *cannot* create a package for a group of servers. However, you can create a package for a server that is in a group.

In the Server Explorer window:

- 1** From the Navigation pane, select Devices and then select All Managed Servers.
- 2** From the Content pane, select a managed server and open it.
- 3** From the Server Explorer, select **Actions > Create Package**.

Or

Select a server object in the object tree, select an object in the content area, and then select **Actions > Create Package**. If the object does not exist in the Software Repository, you will be prompted to provide the source.

Creating a Package from Audit Results

You can create a package from audit results by using the following different windows in the SAS Client.

In the Server Explorer window:

- 1** From the Navigation pane, select Servers and then select All Managed Servers.
- 2** From the Content pane, select a managed server and open it.
- 3** From the Server Explorer, select Audit and Remediation.
- 4** From the Content pane, select the Audit Results tab.
- 5** Select an audit and then select **Actions > Create Package**.

From the Library:

- 1** From the Navigation pane, select Library and then select Audit and Remediation.
- 2** Select Audit Results, and then from the Contents pane select an audit result.
- 3** Right-click the audit result and select **Create Package**.

Creating a Package from a Snapshot

You can create a package from a snapshot by using the following different windows in the SAS Client.

In the Server Explorer window:

- 1** From the Navigation pane, select Servers and then select All Managed Servers.
- 2** From the Content pane, select a managed server and open it.
- 3** From the Server Explorer, select Audit and Remediation.
- 4** From the View drop-down list, select Snapshots.
- 5** From the Content pane, select a snapshot and then select **Actions** ► **Create Package**.

In the Snapshot window:

- 1** From the Navigation pane, select Library and then select Audit and Remediation.
- 2** Select Snapshots, and then from the Content pane, select a snapshot and double-click it to open it.
- 3** From the Snapshots Browser window, select **Actions** ► **Package Snapshot Results**.

Creating a Package

You can create installable packages in native formats, such as Solaris Package and MSI. When you create an installable package, you must specify where you want it uploaded in the Library. You can create a new software policy or add the package to an existing software policy in the Library. When you add a package to an existing software policy in the Library, the contents of your new software policy will overwrite the contents of the existing software policy. See the *Opware[®] SAS Policy Setter's Guide* for information about how to upload packages to and manage software policies.

Optionally, you can specify the operating system versions that you want this package installed on.

In the SAS Client, detaching a software policy containing a Windows zip package created using Visual Packager and then remediating the server does not remove the Windows zip package completely from the server. This behavior is observed if the directory containing the zip package is in use by some other applications.

Additionally, If you try to create a package selecting “all” of a category, such as All COM+ objects, All IIS Metabase, the entire registry, or even the entire file system of a server, you may not be able to successfully create the package because of file size limitations.

To create a package, perform the following steps:

- 1** From one of the starting points described in “Ways to Create a Package”, select Create Package.



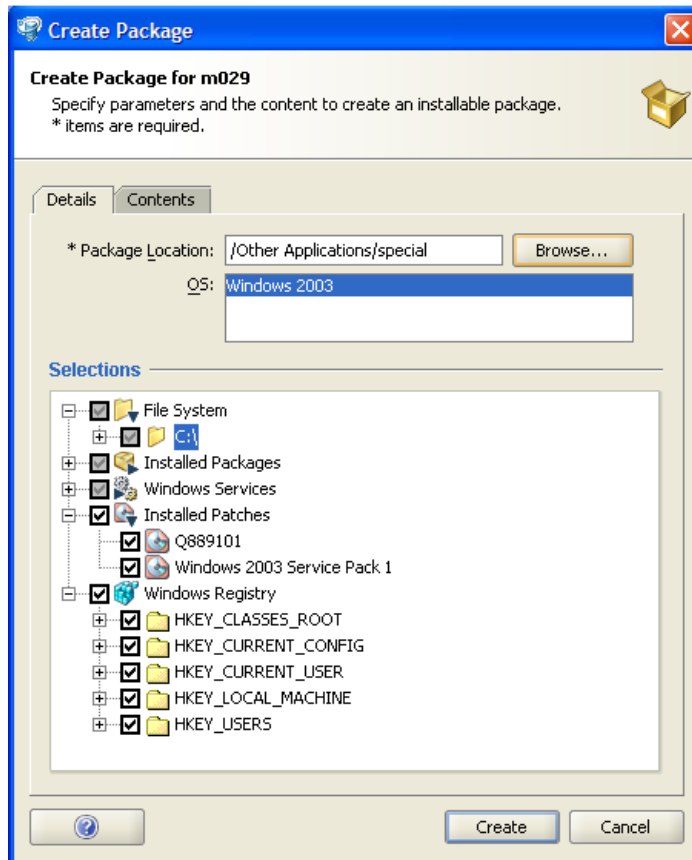
If you have not already set up your packaging server, a warning icon displays in the Create Package window. See “Packaging Server Setup” on page 248 in Chapter 5 for more information.



You must have a set of permissions to manage selection criteria. To obtain these permissions, contact your Opware administrator. See the *Opware® SAS Administration Guide* for more information.

- 2 In the Create Package window, select the Details tab.

Figure 5-3: Sample Create Package Window (Details Tab)



- 3 Next to the Package Location field, click **Browse**. This is a required field, as indicated by the asterisk (*). The Specify Package Location window appears.
- 4 In the Specify Package Location window, specify the location in the folder where you want to create a software policy by selecting a folder from the hierarchy and then entering the name of the software policy in the Package Location field.

The backslash character (/) denotes a new hierarchy.

Click **Apply** to add the new Software Policy.



If you enter a new software policy name that is identical to a name that already exists in the folder hierarchy, the contents of your new software policy will overwrite the contents of the existing software policy.

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

- 5** In the OS field, select one or more operating system versions that you want to install this package on. The operating systems in this list represent the operating system family that you previously configured as a packaging server. You can only select from this list of operating system versions. See “Configuring Options for a Packaging Server” on page 251 in this chapter for more information.



If you select more than one operating system, then the name of software policy containing the package is identical to the operating system selected. You can rename the software policy after creating the package using Visual Packager.

- 6** In the Selections field, select the server objects that you want to include in your package, such as File System, Installed Packages, Installed Patches, Windows Services, and so on. If you select installed packages or installed patches that need to be added to the Software Repository, Visual Packager will instruct you to select the Contents tab to perform this task. See “Adding New Package Content” on page 261 in this chapter for more information.



In Visual Packager, if you create a package from a managed server or audit results with Installed Patches as the selection criteria, and if the Windows patch record does not exist in the patch database, the package is not created.



For Windows Services, you are selecting only the state of the service, such as Started, Stopped, Paused, and so on.

- 7** Click **Create** to create the package and save it in a new or existing software policy in the folder hierarchy, or click **Cancel** to close this window without creating a package.



In Visual Packager, you can create a software policy containing filesystem, Windows Registry, or Windows services. Any changes made to the filesystem, Windows Registry, or Windows services in the software policy are not reverted when you detach the software policy from the server and then remediate the server.

Adding New Package Content

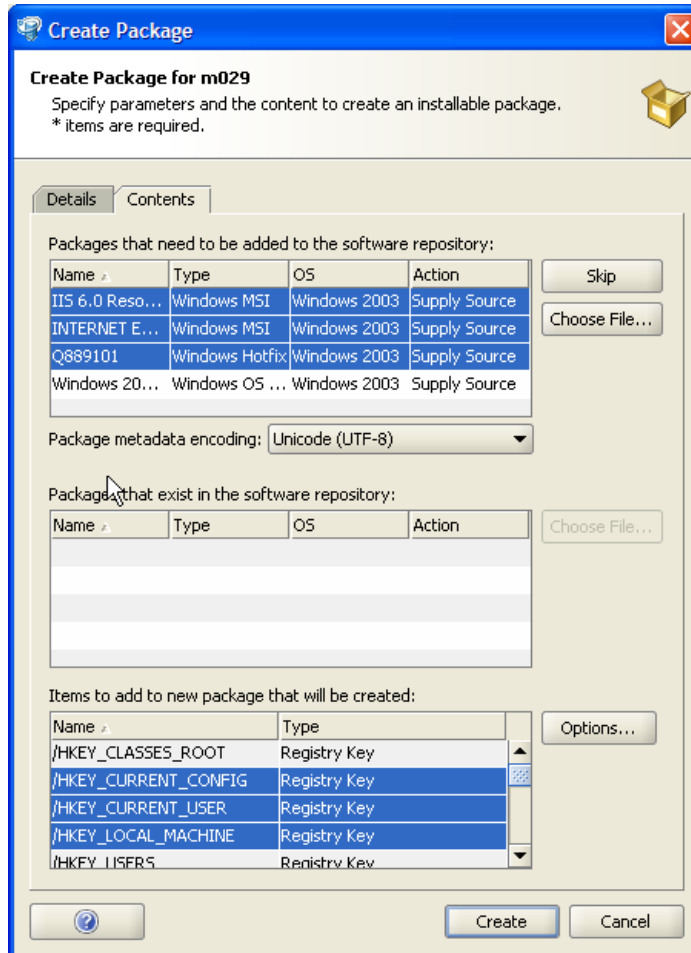
After you have selected packages, patches, or non-package content, Create Package examines them to determine if additional information is required. At this point, Create Package also allows you to pick and choose which packages you want to include or exclude, and which non-package content objects will be packaged.

To add new package content, perform the following steps:

- 1** From one of the starting points described in “Ways to Create a Package”, select Create Package.

- 2** In the Create Package window, select the Contents tab.

Figure 5-4: Sample Create Package Window (Contents Tab)



- 3** In the “Packages that need to be added to the software repository” section, a list of packages or patches (previously selected in the Details tab) that the Software Repository does not have yet is displayed. Create Package requires that you specify the source location of these packages or patches so that they will be uploaded to the Software Repository. Without this source information, Create Package cannot create a package. The following information about these packages is displayed:

- **Name:** The name of the package.
- **Type:** The type of package, such as Sol_Pkg, MIS, RPM, and so on.
- **OS:** The operating system the package is intended for.

- **Action:** If the action status is Supply Source, you must specify the location of the package so that it will be included in the upload process.

You can pick and choose which packages and patches you want to provide source information for by performing the following steps:

1. To exclude particular packages from the upload process, select one or more packages and click **Skip** to exclude them from the upload process. These packages will *not* be attached to the software policy.
2. To include a package in the upload process, select a package and click **Choose File** to locate it in your local file system or in the network file system of the packaging server. This package will be attached to the software policy.

You can also specify the character encoding for the metadata of Unix packages that are displayed in the “Packages that need to be added to the software repository” section. Package metadata includes comments, READMEs, scripts, descriptions, and content lists. Before storing the package internally, Opware SAS translates the metadata from the specified encoding into UTF-8. The default encoding is determined by the source of the package content, either a managed server, server snapshot, or audit results.

To specify character encoding for Unix package metadata, from the “Package metadata encoding” list, select a character encoding, such as Shift-JIS for Japanese. (For Windows packages, the encoding is set to UTF-8.)

- 4 In the “Packages that exist in the software repository” section, a list of packages or patches (previously selected in the Details tab) that currently exist in the Software Repository is displayed. If you want to overwrite these packages and patches with different versions, you must provide the source locations of them. Different package and patch versions are typically ones that you have previously used (and therefore know) are the ones you need to include in the package.

The following information about these packages is displayed:

- **Name:** The name of the package as it appears in the Software Policy.
- **Type:** The type of package as it exists in the Software Policy, such as Sol_Pkg, MIS, RPM, and so on.
- **OS:** The operating system the package is intended for, such as SunOS 5.8.
- **Action:** If the action status is Supply Source, you must specify the location of the package so that it will be included in the upload process.

To choose a package or patch that you want to overwrite the version in the Software Repository, select one and then click **Choose File** to point to its storage location in your local file system or in the network file system of the packaging server.

- 5** In the “Items to add to new package that will be created” section, a list of non-package content (previously selected in the Details tab) is displayed. By default, Create Package creates a new package that contains all of these items. You can specify a separate set of options that apply only to these non-package content items, such as reboot requirements and pre/post install and pre/post uninstall scripts. This non-package content is considered to be a package that is part of the same software policy that the other packages (packages and patches) are attached to.

The following information about these packages is displayed:

- **Name:** The name of the object, such as a directory, file, Windows Registry, Windows Services, and so on.
- **Type:** The type of object.

Click **Options** to display the New Package Options window where you can specify additional settings for all items in the new package content. Visual Packager enables **Options** only if there is additional package content. See “Specifying Options for New Package Content” on page 264 in this chapter for more information.

- 6** Click **Create** to attach this new package content to the software policy, or click **Cancel** to close this window without attaching any new package content to the software policy.

Specifying Options for New Package Content

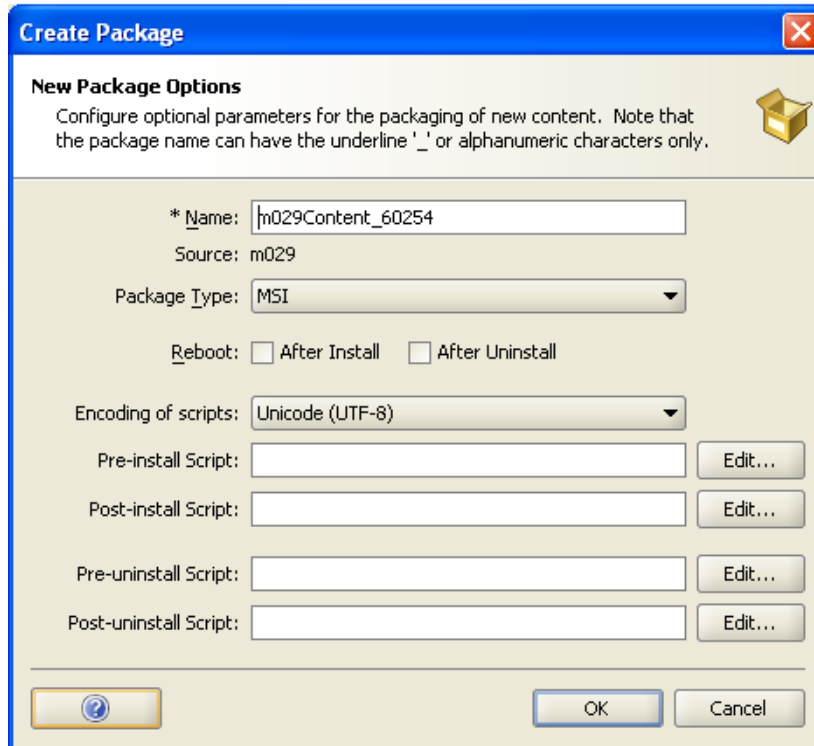
When you add new package content to a package that you have already created, you can specify additional settings that apply only to the new package content. These settings include whether to reboot after your package is installed or uninstalled, which pre/post install and pre/post uninstall scripts will be run, and so on.

To specify options for new package content, perform the following steps:

- 1** Follow the instructions in step 5 in “Adding New Package Content” on page 261 to display the New Package Options window.

- 2** In the New Package Options window, select or enter your preferred options.

Figure 5-5: Sample New Package Options Window



- 3** In the Name field, Visual Packager presets the name of the package. However, if required, you can carefully modify the package name. The following rules apply to package names:
- For Solaris packages, the package name must not exceed five characters, such as 123_S.
 - For all other types of packages, the package name must not exceed 64 characters, such as m123acmecomContent_89787. These types of package names are derived from the source name with the text Content and a 5-digit timestamp appended to it.
 - For all package names, alphanumeric characters and the underscore () character are valid.

This is a required field, as indicated by the asterisk (*).

- 4 The Source field identifies the name of the source of the package. A package source can be a managed server, a snapshot, or an audit result. You cannot modify this field.
- 5 In the Package Type drop-down list, select the format of the package to be created by the Visual Packager. For example, you can select RPM or ZIP for a Linux server. The ZIP format is supported on all platforms.



You must have RPM installed on your packaging server to enable the Visual Packager feature to create an RPM package for Solaris and AIX. The Visual Packager feature does not verify whether RPM is available on the packaging server.

- 6 In the Reboot field, perform any of the following steps:
 - Check After Install to specify that the server will be rebooted after your package is installed. The default is an unchecked reboot option.
 - Check After Uninstall to specify that the server will be rebooted after your package is uninstalled. The default is an unchecked reboot option.

Reboot settings are saved in the Model Repository.

- 7 In the “Encoding of scripts” list, select a character encoding, such as Shift-JIS for Japanese.

This encoding applies to the contents of the pre/post install and pre/post uninstall scripts when they are deployed on a managed server. The default encoding is determined by the source of the package content. Before installing the package on a managed server, Opware SAS converts the scripts into the specified encoding. Internally, Opware SAS stores the scripts in UTF-8.

For Linux, if a script contains non-ASCII characters, then include the shell execution command (such as `#!/bin/sh`) at the beginning of the script.

- 8 For non-ZIP packages, in pre/post install and pre/post uninstall scripts section, you can perform any of the following actions:
 - To specify that a script is run before the package is installed, enter the Pre-install Script or click **Edit** to enter the script in the Specify Post-install Script window.
 - To specify that a script is run after the package is installed, enter the Post-install Script or click **Edit** to enter the script in the Specify Post-install Script window.

- To specify that a script is run before the package is uninstalled, enter the Pre-uninstall Script or click **Edit** to enter your script in the Specify Pre-uninstall Script window.
- To specify that a script is run after the package is uninstalled, enter the Post-uninstall Script or click **Edit** to enter your script in the Specify Post-uninstall Script window.

For ZIP packages, entries in the fields described in this step are ignored. Pre/post install and pre/post uninstall scripts are stored in the Model Repository.



Create Package cannot verify that a script will successfully execute. Be sure to test all scripts before you add them as package options.

- 9 Click **OK** to save your new package options, or click **Cancel** to close this window without saving your new package options.

Viewing Package Details

You can use the Job Manager in the SAS Client to review the following information about your package process:

- Job ID, start and end time, the name of the packaging server, the status of the packaging process, and so on.
- Log details about the package creation process, such as error descriptions.

Appendix A: CML Tutorial and Reference

IN THIS APPENDIX

This section contains the following topics:

- About the CML Tutorial
- CML Fundamentals
- Creating a CML Template
- Completed url_scan_ini.tpl Template
- Using DTD Tags in CML
- Sequence Aggregation
- CML Grammar
- CML Options

About the CML Tutorial

This guide shows you how to use Opsware's Configuration Markup Language (CML) to make an Opsware Application Configuration Template based upon the Microsoft Internet Information Services (IIS) Web server configuration file UrlScan.ini.

While this tutorial will not teach you everything there possibly is to know about CML, creating a CML template from UrlScan.ini will help you gain both a fundamental understanding of CML and the process of creating an Application Configuration Template from a real configuration file.

For more information on how to use Application Configurations in the SAS Client to manage your applications in your managed server environment, see the *Opsware® SAS User's Guide*.

CML Fundamentals

This section contains the main terms and concepts you will need to be familiar with in order to understand this tutorial.

- What is an Application Configuration Template?
- What is an Application Configuration?
- What is CML?
- About the CML Parser
- Anatomy of a CML Tag
- CML Tags You Should Know

What is an Application Configuration Template?

An application configuration template is a “templated” version of an actual configuration file whose values have been turned into variables. Using the SAS Client, a user can edit a template’s value sets and propagate those changes to an actual configuration file on a server.

Once the template version of the configuration file has been created and added to an Application Configuration (inside the SAS Client user interface), system administrators can easily define values for configuration files on servers and server groups.



A CML template is a text file that uses the TPL extension. The Application Configuration feature accepts non-ascii characters, but all key names in your CML templates must be in ASCII. Other fields and text can be either ASCII or non ASCII text.

What is an Application Configuration?

An Application Configuration is like a container or folder which houses Application Configuration Templates. If an application contains several configuration files, you can create an Application Configuration Template for each configuration file you want to manage, and then create a single Application Configuration to contain all the templates. In addition to housing Application Configuration Templates, an Application Configuration can also contain Pre and Post-install scripts that can be executed before and after a configuration push.

What is CML?

CML (Configuration Markup Language) is a configuration markup language that allows system administrators to “templatize” or variablize entries in a native configuration file so those files can be edited and managed from a single location inside the SAS Client.

CML uses special markup tags to modify an application’s configuration’s file’s content – data such as directives, definitions, and so on – so that the configuration data becomes transformed into variables. Once the configuration file has been templated and added to an Application Configuration inside the SAS Client, the end user can manage, edit, and make changes to the native configuration files on managed servers.

About the CML Parser

The CML parser is the engine that utilizes CML configuration files to extract values from existing template files, where they can then be edited. The parser also uses the same CML configuration files to regenerate those configuration files, with new values. For this tutorial, you do not need to know the technical details of the CML parser. You do need to know that CML is used to represent the structure and format of a configuration file, allowing editing of live configuration files on managed servers.

Anatomy of a CML Tag

The basic structure, or anatomy, of a CML tag follows this structure:

```
@<level><tag type><name>;<data type>;<range>;<option1>;<option2>;(more options)@
```

At its most fundamental level, each CML tag starts and end with the @ symbol.

Everything else between consists of one or more of the following CML elements:

- level
- tag type
- name
- data type
- range
- options

level

Levels are used to nest blocks within other blocks. (A block defines parsing rules for a section of information inside a native configuration file.) Levels also signify whether the block spans multiple lines or just one line. Levels under 100 span multiple lines; levels over 100 are contained in a single line. Blocks that start at a level less than or equal to the surrounding block close that block, so when you are nesting blocks use block levels higher than the encompassing block.

You would want to use nested blocks to specify how levels will exist in a block of information in the template. For example, if you had a section in a configuration file that contained two types of data, such as ordered lines and unordered line, then you could set two or more nested blocks (at different levels) in order to handle the information in the section.

tag type

Indicates the type of tag, denoted by a symbol, such as: comment (#), loop: (*), loop target (.), instruction (!), and so on.

name

Indicates where the data read by this tag is stored in namespace. Names can be either absolute or relative. For example:

- **Relative:** If you defined `@!namespace="/system/service/webserver/"@`, then the relative name `@ListenPort@` would use namespace `/system/service/webserver/ListenPort`. Subsequently, you could then use the relative name `@LogFile@` and this name would use the namespace `/system/service/webserver/LogFile`.
- **Absolute:** For example, `@/system/service/webserver/ListenPort@`.

If you define an absolute namespace in the header of a template, then all relative instances of a name will be appended to that namespace. Conversely, you can define an absolute name any time you wish inside the template, and a new namespace will be created.

data type

Indicates what data type the tag will handle. Basic data types can be string, int, decimal, hostname, ordered-string-set, unordered-int-set, ordered-hostname-list, and so on.

range

Specifies ranges of data. For example, you could use a range specifier to specific data range for an int type `1 > 2000`, or the range for a string type: "Windows", "Linux" and so on.

options

Options are very similar to instruction tags and serve to modify or affect the behavior of the tag. They can also be appended to the end most tags, separated by semicolons. For example:

```
;delimiter-is-comma;optional;boolean=yes-format=yes;boolean-no-format=no;
```

CML Tags You Should Know

In order to create a CML template for `UrlScan.ini`, you should become familiar with the following CML tags:

- Comment Tag
- Instruction Tag
- Replace Tag
- Block Tag
- Loop Tag
- Loop Target Tag

Comment Tag

The comment tag can be used to insert information about the template, the configuration file it represents, template metadata (creator, applicable systems, etc.) or any other human-readable information.

Syntax

```
@# <one line comment>EOL
```

Or

```
@## <comments spanning multiple lines> #@
```

The comment tag is often used at the beginning of a CML template (the header) so the author can provide information about the template, such as the name of the template, the configuration file the template as based upon, the purpose of the template, a description of the template, the author, the date, and so on.

Instruction Tag

The instruction tag sets options that will be used at parse time. For example, defining the namespace, whether a list is sorted, ordered, or unordered, how the parser should interpret white space, acceptable delimiters, defining comment characters, and so on.

Syntax

```
@! [{options}]@
```

Replace Tag

The replace tag functions to replace the tag in a CML line with the data from that location in namespace. It is an indicator that the text in this location is data, and it also specifies details about how that data should be stored and validated.

Syntax

```
@{source} [; [{type}] [; [{range} [; {option} [; {option}] ...]]]@
```

The only required element in a replace tag is the source; everything else is optional.

Block Tag

The block tag allows you to group related configuration statements. A block defines parsing rules for a section of information inside a native configuration file. For example, you might have a section of a configuration file where true/false values are defined as either 1/0. In another section in the same file true/false values are set to T/F. You could use the use the block tag to separate the two different ways the CML parser interprets these different ways of defining true/false.

Another example would be if in one section of a configuration file a specific number of spaces are important, while in another section any number of spaces is acceptable, you would use the block tag to indicate where the configuration statements differ.

Syntax

```
@[{level}] [; {option} [; {option}] ...]]]@ <block> <explicit  
or implicit block end>
```

Loop Tag

The loop tag allows sequences (lists and sets) to be enumerated. The block associated with a loop element will be processed for each incident of that block in an input file, and will be generated in an output file for each incidence of that data in a valueset.

Syntax

```
@[{{level}}]*{source} [; [{{type}}] [; [{{range}} [; {option} [; {option}]
...]]]]@ <block> <explicit or implicit block end>
```

Loop Target Tag

The loop target tag is used in a block that is encapsulated by a sequence or type other than namespace. When encountered in a block, this tag is simply replaced with the value of the current value with each loop iteration.

Syntax

```
@. @
```

Creating a CML Template

This section shows you how to create an Application Configuration using CML and contains the following sections:

- Materials Needed for the Tutorial
- Completed Template Sample

Materials Needed for the Tutorial

- Documentation for UrlScan.ini
- UrlScan.ini file
- A text editor

Completed Template Sample

After you finish this tutorial, you can look at the completed url_scan_ini.tpl template so you can compare your results with a completed template. To view the completed template, see “Completed url_scan_ini.tpl Template” on page 297.

1. Read Native Configuration File and Documentation

Once you have identified an application configuration file you want to manage with ACM, the first thing to do is to analyze the native configuration file and its documentation. Make sure that you understand the purpose of the configuration file and all the elements

For example, the documentation for UrlScan.ini tells you that the configuration file enables systems administrators to configure IIS to screen and analyze HTTP requests in order to prevent Internet attacks.

UrlScan.ini consists of several sections, such as [Options], [AllowVerbs], [DenyVerbs], [DenyHeaders], [AllowExtensions], and [DenyExtensions]. Each section allows you to set different configurations to either allow or not allow certain kinds of HTTP requests on your IIS Server.

Each of these sections, judging from the documenting, do not need to be arranged in any specific order. For example, I could list the [Options] sections followed by [DenyVerbs] instead of [AllowVerbs], and the file would still contain the same configuration information and perform its function within IIS. In other words, the order of the main sections of the configuration file is not important

However, the information inside each of these sections do need to be listed (ordered) in a specific way. In other words, the [AllowVerbs] section must be followed by specific verbs you do not want to allow to access your web site. For example, if you put the actual verbs before the [AllowVerbs] string, then that feature of the configuration file would not work.

You also want to become familiar with the kinds of data the configuration file manages. In general, UrlScan.ini works with lists of strings, such as lists of verbs and file extensions. In addition, the file also allows the user to set several yes or no (boolean) options. This kind of information is useful to know before you start creating an Application Configuration Template.

2. Create CML Template File for UrlScan.ini

A CML Template begins as a simple text file that uses the TPL extension.

To create the UrlScan.ini template:

- 1** Using a text editor, create a new text file and save it as Url_Scan_ini.tpl. TPL is the file extension used by Opware for CML templates, though technically you can use any file extension you want, or none at all. Opware Application Configuration Template file naming conventions typically uses the name of the native configuration file with underscores between each section of the native configuration filename.
- 2** Now that you have created the CML template file, you are now ready to build the basic structure of the template, which will consist of a Header, Basic Setup Section, and Template Body.

3. Create CML Template Header

The purpose of creating the CML template header is so that anyone who reads this template will know:

- Name of the native file this template manages (file's absolute pathname)

- Operating systems that the file can work on
- Version of the template
- Author of the template (optional: author's email address)

The first CML tag you will use to create the template's header will be the Comment tag, which allows you to write information about the template. The Comment tag uses this syntax:

```
@# <one line comment>EOL
Or
### <comments spanning multiple lines> #@
```

To create the CML template header, using the CML Comment tag, create a header section at the top of the file that contains three lines of content, the native configuration file that the template will manage, the template version, and the author (with email address).

For example, here's what your template header might look like:

```
@#####
# #
# \system32\inetsrv\urlscan.ini (Windows) #
# Version 1.0 #
# Joe Author (joe_author@your_company.com) #
# #
#####@
```

4. Create CML Template Basic Setup Section

This basic setup section is where you list CML options that instruct the parser how to interpret the CML file. This section can include such as namespace definition, white space handling, list rules, line rules, and so on.

To create the CML template basic setup section:

- 1 Following the header of the CML template, enter the following information (or copy and paste from here):

```
@!namespace=/security/@
@!filename-key="/test";filename-default="/c/UrlScan.ini"@
@!optional-whitespace@
@!boolean-yes-format="1";boolean-no-format="0"@
```

```
@!line-comment-is-semicolon@
@!unordered-lines@
```

This information defines important rules for the url_scan_ini.tpl template, indicating how the CML parser is supposed to interpret and handle information in the template. Notice that each line is a CML instruction tag. You know this is a CML instruction tag because the way the tag starts:

```
@!
```

with an at (@) sign and an exclamation mark (!).

CML Template Basic Setup Section Explained

Table A-1 explains what each section of the template basic setup section means and does.

Table A-1: CML Template Basic Setup Section Explained

CML TAG	DESCRIPTION
@!namespace=/security/@	Define the namespace; in other words, this defines where in the Opware Model Repository values read by the CML template will be stored.
@!filename-key="/files/urlscan_ini";filename-default="/c/urlscan.ini"@	<p>filename-key Defines the location in namespace where the filename will stored.</p> <p>filename-default Defines the location where the native configuration file will be saved on the disk. This path can be changed by the user from the SAS Client.</p> <p>Note that the path names use only forward slashes.</p>

Table A-1: CML Template Basic Setup Section Explained (continued)

CML TAG	DESCRIPTION
<code>@!optional-whitespace@</code>	Indicates that whitespace is optional between items in the configuration file. For example, either of the following entries would be valid if this option is set: Key = "value" Key="value"
<code>@!boolean-yes-format="1";boolean-no-format="0"@</code>	Defines the allowable boolean values in the configuration file. In this case, Yes is indicated with the character 1, and No is indicated with a 0. This means that if a user tried to use the string <code>yes</code> , the Application Configuration would not accept it.
<code>@!line-comment-is-semicolon@</code>	Instructs the parser not to read anything that follows a semicolon in the configuration file. This allows an end user to make comments in the native configuration file using the semicolon before each comment.
<code>@!unordered-lines@</code>	Tells the parser that the sections in the configuration file can be in any order. If you used <code>ordered-lines</code> , then the configuration file would have to conform to the order of the template.

5. Create Template Body

Now that you have created both the header and basic setup portions of your CML template, you are now ready to construct the body. The body is where all your main instructions will be contained.

To create the template body:

- 1 The first thing to do is create a heading that indicates to anyone who might read this file that this is the beginning of the body of the template. Enter the following at the end of the basic setup section of the template:

```
@#####  
# Begin data #  
#####
```

- 2 Save the changes to the file.

6. Mark Up UrlScan [Options] Section – Opening Blocks

Now you are ready to start marking up the template. The first section of the UrlScan.ini file you will convert into CML is the [Options] section, which contains several options for the configuration file.

In CML, if a section of information in a configuration file has more than one kind of data (data that needs to be read differently by the CML parser), you can open “blocks” to handle each section of information separately. Typically, you open a block in CML in order to define special parser rules for a section of the CML file. In the case of the [Options] section, there are basically two “blocks” of information that need to be read by the CML parser: the title of the section and all the options. Since both of these blocks belong together, you will set them at different levels, the first block (the title of the section) at level one, and the second block (the contents of the section) at level two. Nesting the blocks in this manner keeps the sections within the block together when read by the parser.

To markup the UrlScan.ini [Options] section:

- 1 After the “begin data” section of the template, enter the following:

```
@1 [;optional;ordered-lines@  
[Options]  
@2 [;unordered-lines@
```

- 2 In the UrlScan.ini file the [Options] section contains a list of key value pairs. We will use the block tag ([]) set at two levels because there are two kinds of data in this section: a heading and followed by a list of key value pairs. The first level block handles the text string “[Options]” while the second level block will handle all of the key value pairs in that section.

Table A-2 explains how to open two block levels for the [Options] section.

Table A-2: Marking Up the Start of the [Options] Section

CML TAG	DESCRIPTION
<code>@1 [;optional ;ordered-lines@</code>	<p>The number 1 sets the first level of the multiline block.</p> <p>[CML block symbol opens a new block.</p> <p><code>optional</code> Indicates that this entire block is optional and not required to be in the configuration file for the file to be "correct".</p> <p><code>ordered-lines</code> Indicates that whatever follows this tag (the string [Options]) has to come first in the native UrlScan.ini configuration file. In other words, you could not list in the native file all the options and then the title. "[Options]" has to come first. In CML, the option "<code>ordered-lines</code>" determines this order.</p>
<code>[Options]</code>	<p>The string that names the section in the native configuration file.</p>
<code>@2 [;unordered-lines@</code>	<p>The number 2 sets the second level of the block.</p> <p>[CML block symbol opens a new block.</p> <p><code>unordered lines</code> Indicates that all the lines that follow [Options] within the block can be in any order in the configuration file. In other words, all the key value pairs that are contained in the [Options] section can be ordered and will be read by parser.</p>

- 3** Next, you will markup all the options lines from the configuration file. Most of these entries use the CML replace tag because they are simply key value pairs that allow a user to replace a single value. Table A-3 explains the CML markup of each option.

Table A-3: Marked Up Key Value Pairs from UrlScan.ini [Options] Section

CML TAG	DESCRIPTION
<pre>AllowDotInPath = @allow_dot_in_path;boolean@</pre>	<p>Note: All of the key value pair markup use some variation of the following syntax (unless otherwise indicated):</p> <pre>string literal = @source;type@ allow_dot_in_path</pre> <p>This string defines the namespace path to store this value. In this example, the namespace is relative, which means that it will be appended to the namespace that you defined in the header of the template (@!namespace=/security/@) and will store the value in that namespace location.</p> <p>For example:</p> <pre>/security/allow_dot_in_path.</pre> <p>If you wanted, you could also write this tag like this:</p> <pre>AllowDotInPath = @/security/allow_dot_in_path;boolean@ boolean</pre> <p>Since the key value pair type is boolean, we used the CML type: <code>boolean</code>. Note that since in the header of this template we defined an acceptable boolean yes value as 1, when the end user modifies the template in the SAS Client, they would need to enter a one if they want to allow dots in the path of IIS.</p>

Table A-3: Marked Up Key Value Pairs from `UrlScan.ini [Options]` Section (continued)

CML TAG	DESCRIPTION
<code>AllowHighBitCharacters = @allow_high_bit_characters;boolean@</code>	Allows users to choose whether or not high bit characters are acceptable in a URL, flagged by a yes (1) or no (2) in the configuration file.
<code>AllowLateScanning = @allow_late_scanning;boolean@</code>	Allows users to choose whether or not late scanning of a URL is acceptable. And, defines a namespace location to store value. <code>boolean</code> indicates this key is accepts a yes (1) or no (2) in the configuration file.
<code>AlternateServerName = @alternate_servername@</code>	Defines a namespace where an alternate server name can be stored when entered by the user, or read in from a configuration file.
<code>EnableLogging = @enable_logging;boolean@</code>	Allows users to turn on logging, flagged by a yes (1) or no (2) in the configuration file.
<code>LoggingDirectory = @logging_directory;dir@</code>	Allows users to choose a directory to store log files, if logging has been turned on. Notice that for the type, the CML tag uses the element <code>dir</code> - an acceptable CML data type.
<code>LogLongURLs = @log_long_urls;boolean@</code>	Allows user to choose whether or not to log URLs that access the server, a yes (1) or no (2) in the configuration file.
<code>NormalizeUrlBeforeScan = @normalize_url_before_scan;boolean@</code>	Allows users to choose whether or not to normalize the URL before it is read by the server, flagged by a yes (1) or no (2) in the configuration file.
<code>PerDayLogging = @per_day_logging;boolean@</code>	Allows users to choose to turn on per day logging, flagged by a yes (1) or no (2) in the configuration file.

Table A-3: Marked Up Key Value Pairs from UrlScan.ini [Options] Section (continued)

CML TAG	DESCRIPTION
<pre>PerProcessLogging = @per_ process_logging;boolean@</pre>	<p>Allows users to turn on or off per process logging, flagged by a yes (1) or no (2) in the configuration file.</p>
<pre>RejectResponseUrl = @reject_response_ url;string;r'(HTTP_URLSCAN_ STATUS_HEADER) (HTTP_URLSCAN_ ORIGINAL_VERB) (HTTP_URLSCAN_ ORIGINAL_URL) ';optional@</pre>	<p>Syntax</p> <p>string literal = @source;type;r'regular expression';option@</p> <p>reject response String literal that defines the path where the strings will be stored in namespace.</p> <p>string Indicates that the data type for the reject URL request is a string.</p> <p>r' A string range specifier that introduces a regular expression. In this case, a range of string literals.</p> <p>(HTTP_URLSCAN_STATUS_ HEADER) (HTTP_URLSCAN_ ORIGINAL_VERB) (HTTP_URLSCAN_ ORIGINAL_URL) '</p> <p>The string literals (rejected URL responses) to be read by the parser: the status header, original verb, and original URL.</p> <p>optional Indicates that this value is optional. That is, if left blank, the parser can still read the CML.</p>

Table A-3: Marked Up Key Value Pairs from `UrlScan.ini [Options]` Section (continued)

CML TAG	DESCRIPTION
<code>RemoveServerHeader = @remove_server_header;boolean@</code>	Allows users to turn on or off the RemoveServerHeading feature. When activated (set to 1), the reject response sent to the client will removing the server header in the message. This setting is flagged by a yes (1) or no (2) in the configuration file.
<code>UseAllowVerbs = @use_allow_verbs;boolean@</code>	Allows users to turn on or off the UseAllowVerbs feature. When activated (set to 1), the server will reject any request to the server that contain an HTTP verb that is not explicitly listed in the AllowVerbs section of the <code>UrlScan.ini</code> file. Flagged by a yes (1) or no (2) in the configuration file.
<code>UseAllowExtensions = @use_allow_extensions;boolean@</code>	Allows users to turn on or off the UseAllowExtension feature. When activated (set to 1), the server will reject any request to the server that contain a file extension that it not explicitly listed in the AllowExtension section of the <code>UrlScan.ini</code> file. Flagged by a yes (1) or no (2) in the configuration file.
<code>UseFastPathReject = @use_fast_path_reject;boolean@</code>	Allows users to turn on or off the UseFastPathReject feature. When activated (set to 1), the server ignores the RejectResponseUrl option and returns a short 404 response to the client when a URL is rejected. Flagged by a yes (1) or no (2) in the configuration file.
<code>VerifyNormalization = @verify_normalization;boolean@</code>	Allows user to turn on or off normalization of all URLs scanned by <code>UrlScan.ini</code> . When activated (set to 1), the URL is normalized before being scanned. Flagged by a yes (1) or no (2) in the configuration file.

7. Closing One Block by Opening a New One

Now that you have marked up all of the options in the [Options] section of the UrlScan.ini file, you are ready to start marking up the next section, [AllowExtensions]. Remember that to start the [Options] section you had to open a two level block to account for two levels of information – the title of the [Options] section and its contents.

Before you can start marking up the [AllowExtensions], you need to close the previous section by closing the CML block. With CML, you can close a block by opening a new block at a higher (lower number) or equal to level. In this task, you will open the new block for the [AllowExtensions] the same way you opened a block for the [Options] section, by starting a new first level block.

To open a new block and mark up the [AllowExtensions] section:

- 1 After the last line of the [Options] section, enter the following text to open the new block for the [AllowExtensions] section:

```
@1 [;optional;ordered-lines@  
[AllowExtensions]  
@2 [;unordered-lines@
```

Table A-4 explains how opening a new two level block closes the previous block.

Table A-4: Starting a New Block for the [AllowExtensions] Section

CML TAG	DESCRIPTION
<pre>@1 [;optional ;ordered-lines@</pre>	<p>The number 1 opens a new level one block. Because it is a number 1 level block, which is at a higher level than the previous block (a level two block for the key value pairs in the [Options] section) and equal to the level 1 block before that, it will close the two blocks that came before it.</p> <p>Note that you could also close a block by using the close block command. For example:</p> <pre>@2]@</pre> <p>[CML block symbol that opens a new block.</p> <p><code>optional</code> Indicates that this entire block is optional and not required to be in the configuration file for the file to be "correct".</p> <p><code>ordered-lines</code> Indicates that whatever follows this tag (the string [AllowExtensions] has to come first in the native UrlScan.ini configuration file. In other words, you could not list all the options in the native file and then the title. [AllowExtensions] has to come first. In CML, the ordered-line element determines this order.</p>
<pre>[Options]</pre>	<p>The literal string that names the section in the native configuration file.</p>

Table A-4: Starting a New Block for the [AllowExtensions] Section (continued)

CML TAG	DESCRIPTION
@2 [;unordered-lines@	<p>The number 2 sets the second level of the block.</p> <p>[</p> <p>CML block symbol that opens a new block.</p> <p>unordered lines</p> <p>Indicates that all the lines that follow [AllowExtensions] within the block can be in any order in the configuration file. In other words, all the key value pairs that are contained in the [AllowExtensions] section can be ordered in any order you wish.</p>

- 2** Next, because the [AllowExtensions] section of the UrlScan.ini file can contain any list of file extensions entered by the user, you will use a CML loop and loop target tag to instruct the parser will read the information in this section one line at a time, then repeat by reading the next line, and so on.

Directly after the last @2 [;unordered-lines@ text from the last step, enter the following text:

```
@*allow_extension;unordered-string-set@
. @ . @
```

Table A-5 explains the how the loop and loop target CML tags work:

Table A-5: Loop and Loop Target CML Tags

CML TAG	DESCRIPTION
<code>@*allow_extension;unordered-string-set@</code>	<p>Syntax</p> <p><code>@<level><tag type><name>;<data type>;<options>@</code></p> <p>The loop tag (*) will “loop” or read over the unordered string set listed in the [AllowExtensions] section.</p> <p><code>allow_extension</code> String that defines the path where the strings will be stored in namespace.</p> <p><code>unordered-string-set</code> Indicates that the list of strings do not have to be listed in any specific order.</p>
<code>.@.@</code>	<p>First (.)</p> <p>In this section, this unordered string set that the parser reads is a list of file extensions listed in the [AllowExtensions] section that start with a (.) character.</p> <p><code>.@.</code></p> <p>Loop target tag (.) instructs the parser to read everything in this list that starts with a period character.</p>

3 Save the file.

8. Mark Up [DenyExtensions] Section by Opening a New Block

In this task, you will markup the [DenyExtensions] section of the UrlScan.ini file the exact same way you marked up the [AllowExtensions] section. You will be opening a new level one block, which closes the previously opened block from the [AllowExtensions] section.

Then, you will open a level two block from which you will instruct the parser to read an unordered list of all file extensions beginning with a (.) that you wish to block using UrlScan.ini.

The CML markup for the [AllowExtensions] section looks like this:

```
@1 [;optional;ordered-lines@
[DenyExtensions]
@2 [;unordered-lines@
@*deny_extension;unordered-string-set@
.@.@
```

9. Mark Up [AllowVerbs] and [DenyVerbs] Sections

The next two sections of the UrlScan.ini file will follow the exact same CML markup as you used for [DenyExtensions] in the previous sections. You will open a first level block to close the previous block, which will also parse the following text as an ordered line.

Then, you will open a second level block that reads the following list of as an unordered strings – in other words, a list of verbs. In these two sections, the string you will instruct CML to read will be a list of verbs you wish to allow into your web site and a list of verbs you wish to deny access to your web site.

The CML markup for both of these sections is as follows:

```
@1 [;optional;ordered-lines@
[AllowVerbs]
@2 [;unordered-lines@
@*allow_verb;unordered-string-set@
.@.@

@1 [;optional;ordered-lines@
[DenyVerbs]
@2 [;unordered-lines@
@*deny_verb;unordered-string-set@
.@.@
```

10. Mark Up [DenyHeaders] Section

In this next task, you will mark up the [DenyHeaders] section of the UrlScan.ini file, which allows you to configure IIS to deny specific HTTP request headers.

This section will be marked up in CML similarly to the previous sections in that you will open two blocks that will be read for strings. However, you will be separating the list of HTTP headers listed in the UrlScan.ini file by a colon, using a CML sequence delimiter. Since HTTP request headers contain a colon (:), you need to use a sequence delimiter to tell the parser to read each line in the section so when it encounters a colon (:), it will move on to the next entry.

For example, the list of HTTP headers to be denied listed in the UrlScan.ini file might read something like this:

```
Translate:
If:
Lock-Token:
```

Because each header request listed in the configuration file ends with a (:), we need to instruct the parser to recognize the (:) as the end of an entry.

To markup the [DenyHeaders] section:

- 1** After the last line of the [DenyVerbs] section, enter the following text to open the new block for the [DenyHeaders] section:

```
@1 [ ;optional;ordered-lines@
[DenyHeaders]
@2 [ ;unordered-lines@
```

As you have done in previous sections, with these tags you are opening a level one block to be read as an ordered line, then opening a second level block to be read as unordered lines.

- 2** Next, type the following CML loop and loop target tags to instruct the parser to read through the list of header requests:

```
@*deny_header;unordered-string-set;;sequence-delimiter=":"@
@.:@
```

Loop and Loop Target Tags for the [DenyHeaders] Section Table A-6 describes the syntax of these two tags.

Table A-6: Loop and Loop Target Tags for the [DenyHeaders] Section

CML TAG	DESCRIPTION
<pre>@*deny_header;unordered- string-set;;sequence- delimiter=":"@</pre>	<p>* Indicates a loop CML tag that will read through the list of strings.</p> <p>deny_header String literal that defines the path where the strings will be stored in namespace.</p> <p>unordered-string-set Indicates that the list of strings can be listed in any order.</p> <p>; The first semicolon separates the two sections of the tag.</p> <p>; The second semicolon allows you to enter the following colon (:) sequence delimiter without it being interpreted as a range.</p> <p>sequence-delimiter=":" Instructs the parser to read a colon (:) as part of the string and the point at which to move on to the next entry.</p>
<pre>@.@"</pre>	<p>Loop target tag instructs the parser to store these values into the <code>deny_header</code> namespace. E.g., <code>/security/deny_extension</code></p>
<pre>:</pre>	<p>Final colon (:) tells the parser that each item in this list is going to be followed by a colon. In other words, this character will be included and stored as a part of the entry for a denied header.</p>

- 3 Save the file.

11. Mark Up [DenyURLSequences] Section

Marking up the [DenyUrlSequence] is very similar to the way in which you marked up the [DenyHeader] section: you will open two blocks that will be read for order and unordered strings. However, for this section you will be separating the list of URL sequences in the template with a field delimiter. The field delimiter used here will be an end of line element (eol) which instructs the parser stop reading an entry when it encounters the end of a line.

To markup the [DenyUrlSequence] section:

- 1 After the last line of the [DenyUrlSequence] section, enter the following text to open the new block for the [DenyUrlSequence] section:

```
@1 [;optional;ordered-lines@
  [DenyUrlSequence]
@2 [;unordered-lines@
```

As you have done in previous sections, with these tags you are opening a level one block to be read as an ordered line, then opening a second level block to be read as unordered lines.

- 2 Next, type the following CML loop and loop target tags to instruct the parser to read through the list of URL sequences to be denied:

```
@*deny_url_sequence;unordered-string-set;;field-delimiter-
is-eol@
@.@
```

Table A-7 describes the syntax of these tags

Table A-7: Loop and Loop Target Tags for the [DenyUrlSequence] Section

CML TAG	DESCRIPTION
<pre>@*deny_url_sequence;unordered- string-set;;field-delimiter- is-eol@</pre>	<p>* Indicates a loop CML tag that will read through the list of strings.</p> <p><code>deny_url_sequence</code> String literal that defines the path where the string will be stored in namespace.</p> <p><code>unordered-string-set</code> Indicates that the list of strings can be listed in any order.</p> <p><code>;</code> The first semicolon separates the two sections of the tag.</p> <p><code>;</code> The second semicolon allows you to enter the following colon (:) sequence delimiter without it being interpreted as a range.</p> <p><code>sequence-delimiter=":"</code> Instructs the parser to read a colon (:) as part of the string and the point at which to move on to the next entry.</p>
<pre>@. @</pre>	<p>Loop target tag instructs the parser to store these values into the <code>deny_url_sequence</code> namespace. E.g., <code>/security/deny_url_sequence</code>.</p>

- 3** Save the file.

12. Mark Up [RequestLimits] Section

Marking up the [RequestLimits] is very similar to the way in which you marked up the [DenyUriSequence] section: you will open two blocks that will be read for order and unordered strings. But for this section, after you open both blocks, you will be using the CML replace tag to mark up three key value pairs.

To markup the [RequestLimits] section:

- 1 After the last line of the [RequestLimits] section, enter the following text to open the new block for the [RequestLimits] section:

```
@1 [;optional;ordered-lines@  
[RequestLimits]  
@2 [;unordered-lines@
```

As you have done in previous sections, with these tags you are opening a level one block to be read as an ordered line, then opening a second level block to be read as unordered lines. Recall that by starting the new first level block, you are closing the previous second level block from the {DenyUriSequence} section.

- 2 Next, type the following CML replace tags to mark up the three key value pairs found in the [RequestLimits] section:

```
MaxAllowedContentLength = @max_allowed_content_length;int@  
MaxUrl = @max_url;int@  
MaxQueryString = @max_query_string;int@  
@1]@
```

Table A-8 describes the syntax of these tags.

Table A-8: Loop and Loop Target Tags for the [DenyUrlSequence] Section

CML TAG	DESCRIPTION
<pre>MaxAllowedContentLength = @max_ allowed_content_length;int@</pre>	<p>MaxAllowedContentLength Request limit parameter string from the configuration file.</p> <p>max_allowed_content_length String literal that defines the path where the value will be stored in namespace.</p> <p>int Indicates that the value to be stored is an integer.</p>
<pre>MaxUrl = @max_url;int@</pre>	<p>MaxUrl Request limit parameter string from the configuration file.</p> <p>max_url String literal that defines the path where the value will be stored in namespace.</p> <p>int Indicates that the value to be stored is an integer.</p>
<pre>MaxQueryString = @max_query_ string;int@</pre>	<p>MaxQueryString Request limit parameter string from the configuration file.</p> <p>max_query_string String literal that defines the path where the value will be stored in namespace.</p> <p>int Indicates that the value to be stored is an integer.</p>
<pre>@1]@</pre>	<p>This level one block tag closes the block.</p>

3 Save the File**13. From Template to Application Configuration**

Once you have completed creating the CML template for UrlScan.ini (saved as url_scan_ini.tpl), you are now ready to do the following tasks:

- Import the template into the SAS Client
- Add the template to an Application Configuration
- Validate the CML syntax
- Attach the Application Configuration to a server
- Test by making changes and pushing changes to the server

For information on how to create an Application Configuration, add a template to it, validate its CML syntax, and attach it to a server, and push changes, see the online help for Application Configuration in the SAS Client.

Completed url_scan_ini.tpl Template

We have included a sample of a completed url_Scan_ini.tpl template so you can compare your work with a finished template.

```
@#####
# #
# \system32\inetsrv\urlscan.ini (Windows) #
# Version 1.0 #
# Joe Author (joe_author@your_company.com) #
# #
#####@

@!namespace=/security/@
@!filename-key="/test";filename-default="/c/UrlScan.ini"@
@!optional-whitespace@
@!boolean-yes-format="1";boolean-no-format="0"@
@!line-comment-is-semicolon@
@!unordered-lines@

@#####
# Begin data #
#####@
```

```
@1[;optional;ordered-lines@
[Options]
@2[;unordered-lines@

AllowDotInPath = @allow_dot_in_path;boolean@

AllowHighBitCharacters = @allow_high_bit_characters;boolean@

AllowLateScanning = @allow_late_scanning;boolean@

AlternateServerName = @alternate_servername@

EnableLogging = @enable_logging;boolean@

LoggingDirectory = @logging_directory;dir@

LogLongURLs = @log_long_urls;boolean@

NormalizeUrlBeforeScan = @normalize_url_before_scan;boolean@

PerDayLogging = @per_day_logging;boolean@

PerProcessLogging = @per_process_logging;boolean@

RejectResponseUrl =
@reject_response_url;string;r'(HTTP_URLSCAN_STATUS_
HEADER) | (HTTP_URLSCAN
_ORIGINAL_VERB) | (HTTP_URLSCAN_ORIGINAL_URL)';optional@

RemoveServerHeader = @remove_server_header;boolean@

UseAllowVerbs = @use_allow_verbs;boolean@

UseAllowExtensions = @use_allow_extensions;boolean@

UseFastPathReject = @use_fast_path_reject;boolean@

VerifyNormalization = @verify_normalization;boolean@

@1[;optional;ordered-lines@
[AllowExtensions]
@2[;unordered-lines@

@*allow_extension;unordered-string-set@
. @. @

@1[;optional;ordered-lines@
```

```
[DenyExtensions]
@2 [;unordered-lines@

@*deny_extension;unordered-string-set@
. @. @

@1 [;optional;ordered-lines@
[AllowVerbs]
@2 [;unordered-lines@

@*allow_verb;unordered-string-set@
@. @

@1 [;optional;ordered-lines@
[DenyVerbs]
@2 [;unordered-lines@

@*deny_verb;unordered-string-set@
@. @

@1 [;optional;ordered-lines@
[DenyHeaders]
@2 [;unordered-lines@

@*deny_header;unordered-string-set;;sequence-delimiter=":"@
@. @:

@1 [;optional;ordered-lines@
[DenyURLSequences]
@2 [;unordered-lines@

@*deny_url_sequence;unordered-string-set;;field-delimiter-is-
eol@
@. @

@1 [;optional;ordered-lines@
[RequestLimits]
@2 [;unordered-lines@

MaxAllowedContentLength = @max_allowed_content_length;int@

MaxUrl = @max_url;int@

MaxQueryString = @max_query_string;int@
@1]@
```

Using DTD Tags in CML

CML supports Document Type Definition (DTD) tags that can be used to pre-define attributes for a CML tag. Using a DTD tag in CML allows you to change some aspects of how the template is displayed in the SAS Client. The DTD definition generally goes in the beginning of a file and the tag gets shortened to just a name and a tag type.

The main advantage of using DTD tags in CML is the ability to define 'printable' and 'description' values, which are reflected in the SAS Client, improving usability. DTD definitions can be used to define any tag that has a name; for example loop tags, loop target tags, replace tags, and so on, but not tags like instruction tags or block tags. DTD tags in CML are also inherently multi-line tags.

DTD Tags Example

Here we will take a tag and create a DTD version of that tag. A DTD tag in CML isn't much different than a regular CML tag; it contains all the elements of a tag minus the "tag type".

For example, in the CML tag below:

```
@*deny_header;unordered-string-set;;sequence-delimiter=":";optional@
```

this is an instance representing the following format in CML:

```
@<tag type><name>;<data type>;<option1>;<option2>@
```

The DTD version of this takes the existing elements and reorders them as follows:

```
<start code block>
@~<name>
type = <data type>
description = <description>
printable = <printable>
<option1>
<option2>
...
@
@<tag type><name>@
<end code block>
```

As you can see, this usage also allows for the addition of two new elements: "description" and "printable". Defining "printable" will define the main text for this tag in the SAS Client. Defining "description" will create a description for this value in the SAS Client that is viewable when the user mouses over the field in the Value Set Editor in the SAS Client.

Here is the same tag in full DTD format:

```

<start code block>
@~deny_header
type = unordered-string-set
printable = Headers to Deny
description = This is a list of headers that IIS should deny
sequence-delimiter = ":"
optional
@
@*deny_header@
<end code block>

```

There are a couple things to notice in the example above. In defining a value for "description," the value can span multiple lines, as long as the lines following the first line have whitespace as the first character.

Options go on a line by themselves, where you have `<option>=<value>` you need to insert spaces before and after the "=" sign.

Now, where ever you use the tag `@*deny_header@`, the parser will use the predefined DTD for all that tags' information.



Redefining a DTD defined tag, `@*deny_header@`, by using a line like `@*deny_header;unordered-string-set@` will cause the CML template to become invalid.



Note also that DTD style CML is not currently required, but is most obvious when viewing the Application Configuration the SAS Client. If you don't use DTD tags you will not see the 'printable' and 'description' fields, instead you will only see the underlying variable name.

Sequence Aggregation

Because Application Configuration values can be set across many different levels in the Application Configuration inheritance hierarchy (also referred to as the inheritance scope), it is important that you be able control the way multiple sequence values are merged together when you push an Application Configuration on to a server.

ACM allows you to control the way sequence values are merged across inheritance scopes. This means that you can, for example, add some values to a sequence in the Customer scope, Group scope, and the Server scope, and all the values will be merged together to form the final sequence.

The manner in which sequence values are merged is controlled by special tags in the CML template, using three different sequence merge modes:

- **Sequence Replace:** Sequence values from more specific scopes completely replace those from less specific scopes. This occurs for both sequences of sets and lists.
- **Sequence Append:** For lists, values at more general scopes are appended (placed after) to those at more specific scopes. Duplicates, if present, are not removed. For sets, the behavior is the same, except duplicates are merged. For lists, duplicates are identified according to child elements marked with the `primary-key` tag, and then merged. For scalars, this is done by simply removing duplicate values, leaving only the value from the most specific scope (the last occurrence is the merged sequence). This is the default mode, and will be used if nothing else is specified.
- **Sequence Prepend:** Works the same as append, but values at more general scopes are prepended (placed before) to those at more specific scopes.

For example, with these two sets:

- “a, b” – At a more specific (inner) level of the inheritance scope, for example, server instance level.
- “c, d” – At a more general (outer) of the inheritance scope, for example, the server group level.

When the application configuration template is pushed onto the server, the merging results would be:

- Sequence replace: “a, b”
- Sequence append: “a, b, c, d”
- Sequence prepend: “c, d, a, b”

Sequence aggregation occurs not only between scopes, but also within a scope itself. This is evident if there are duplicate values within a sequence of namespaces.

Sequence Replace

In the Replace merge mode (CML tag “`sequence-replace`”), the contents of a sequence defined at a particular scope replace those of less specific scopes, and no merging is performed on the individual elements of the sequence.

For example, if the `sequence-replace` tag has been set for a list in an Application Configuration Template CML source, then values set for that list at the server instance level will override, or replace, those set at the group level and at the Application Configuration default values level.

For example, if a list in an `etc/hosts` file was defined at the group level (outer) as the following:

```
/system/dns/host/1/ip           127.0.0.1
/system/dns/host/1/hostnames/1 localhost
/system/dns/host/1/hostnames/2 mymachine
/system/dns/host/2/ip           10.10.10.10
/system/dns/host/2/hostnames/1 loghost
```

And the same list was defined at the device scope (inner), as the following:

```
/system/dns/host/1/ip           127.0.0.1
/system/dns/host/1/hostnames/1 localhost
/system/dns/host/1/hostnames/2 mymachine.mydomain.net
/system/dns/host/2/ip           10.10.10.100
/system/dns/host/2/hostnames/1 mailserver
```

If template had defined the `/system/dns/host` element with the `sequence-replace` tag, the final results of the configuration file on the server after the push would be:

```
127.0.0.1 localhost mymachine.mydomain.net
10.10.10.100 mailserver
```

Sequence Append

When the append list merge mode (CML tag “`sequence-append`”) is used for sequences, the values at more general scopes are appended (placed after) those of more specific scopes. Sequence append mode is the default mode for merging list values. If nothing is specified in the CML of the template, the sequence append will be used.

If a list in an `etc/hosts` file was defined at the group level (outer) as the following:

```
/system/dns/host/1/ip           127.0.0.1
/system/dns/host/1/hostnames/1 localhost
```

```
/system/dns/host/1/hostnames/2 mymachine
/system/dns/host/2/ip          10.10.10.10
/system/dns/host/2/hostnames/1 loghost
```

And the same list was defined at the device scope (inner), as the following:

```
/system/dns/host/1/ip          127.0.0.1
/system/dns/host/1/hostnames/1 localhost
/system/dns/host/1/hostnames/2 mymachine.mydomain.net
/system/dns/host/2/ip          10.10.10.100
/system/dns/host/2/hostnames/1 mailserver
```

Using the value sets from the above example, if the `/system/dns/host` element was a list with the `sequence-append` tag set in the Application Configuration Template, the final results of the configuration file on the server after the push would be:

```
127.0.0.1 localhost mymachine.mydomain.net
10.10.10.100 mailserver
127.0.0.1 localhost mymachine
10.10.10.10 loghost
```

But since it is not allowable for a hosts file to contain duplicate entries, the `/system/dns/host` element will have to be flagged in the Application Configuration Template as a set rather than a list, because sets do not allow duplicates. To avoid duplication of the list values in the example, the Application Configuration Template author would use the Primary Key option.

Primary Key Option in Sequence Merging

When operating in append mode on sets, new values in more specific scopes are appended to those of less specific ones, and duplicate values are merged with the resulting value placed in the resulting sequence according to its position in the more specific scope.

How this affects merged sequence values depends on what kind of data is contained in the sequence:

- For elements in a sequence which are scalars, the value from the most specific scope is used. In other words, values at the server instance level would replace the values at the group level.
- For elements which are namespace sequences, the value is obtained by applying the merge mode specified for that element (in this example, append) based upon matching up the primary fields.

To avoid the duplication of the `/system/dns/host/.ip` value, the Application Configuration Template author would use the CML `primary-key` option. With this option set, ACM will treat entries with the same value for `/system/dns/host/.ip` as the same and merge their contents.

In the example above, the final results of the configuration file on the server after the push would be:

```
127.0.0.1 localhost mymachine.mydomain.net mymachine
10.10.10.100 mailserver
10.10.10.10 loghost
```



Since it is possible to have a set without primary keys, if there are scalars in the sequence, then an aggregation of all scalar values will be used as the primary key. If there are no scalars, then the aggregation of all values in the first sequence will be used as the primary key. Although this is an estimate, in most cases the values will be merged effectively. To ensure that the correct values are used as primary keys, we recommend that you always explicitly set the primary key in a sequence.

Sequence Prepend

When the append list merge mode (CML tag “`sequence-prepend`”) is used for sequences, the values at more general scopes are prepended (placed before) those of more specific scopes.

For example, if a sequence in an `etc/hosts` file was defined at the group level (outer) as the following:

```
/system/dns/host/1/ip          127.0.0.1
/system/dns/host/1/hostnames/1 localhost
/system/dns/host/1/hostnames/2 mymachine
/system/dns/host/2/ip          10.10.10.10
/system/dns/host/2/hostnames/1 loghost
```

And the same sequence was defined at the device scope (inner), as the following:

```
/system/dns/host/1/ip          127.0.0.1
/system/dns/host/1/hostnames/1 localhost
/system/dns/host/1/hostnames/2 mymachine.mydomain.net
/system/dns/host/2/ip          10.10.10.100
/system/dns/host/2/hostnames/1 mailserver
```

If the `/system/dns/host` element was a set with the `sequence-prepend` tag set in the Application Configuration Template, the final results of the configuration file on the server after the push would be:

```
10.10.10.10 loghost
127.0.0.1 mymachine localhost mymachine.mydomain.net
10.10.10.100 mailserver
```

CML Grammar

Table A-9 describes CML grammar illustrating several types of CML tags.



Many elements list in the grammar are not covered in the tutorial.

Table A-9: CML Grammar

CML TAG/ELEMENT	DESCRIPTION
replace-tag	"@" source [";" [type] [";" [range] *option]] "@"
data-definition-tag	"@~" source CRLF *def-line "@"
conditional-tag	"@" [group-level] "?" source [";" [type] [";" [range] *option]] "@"
loop-tag	"@" [group-level] "*" source [";" [type] [";" [range] *option]] "@"
loop-target-tag	"@.@"
block-tag	"@" [group-level] "[" *option "@"
block-termination-tag	"@" [group-level] "]"@"
line-continuation-tag	"@\\"
instruction-tag	"@!" *option "@"
single-line-comment	"@#" string CRLF
multi-line-comment	"@##" *[string / CRLF] "#@"

Table A-9: CML Grammar (continued)

CML TAG/ELEMENT	DESCRIPTION
def-line	type-line / range-line / option-line / printable-line / desc-line
type-line	"type" WSP "=" WSP type-elem CRLF
range-line	"range" WSP "=" WSP range CRLF
option-line	option-elem CRLF
printable-line	"printable" WSP "=" WSP string CRLF
desc-line	"description" WSP "=" *[WSP string CRLF]
group-level	int
source	absolute-path / relative-path / local-path
absolute-path	"/" path-component* name
relative-path	[path-component*] name
path-component	(name / sequence-id) "/"
sequence-id	int
local-path	"." name
name	string
type	sequence / type-elem
sequence	[order "-"] type-elem "-" sequence-elem
sequence-elem	"set" / "list"
type-elem	"int" / "string" / "ip" / "port" / "file" / etc...
order	ordered" / "unordered"
range	and-range *["," and-range]
and-range	range-elem *["&" range-elem]
range-elem	numeric-range / string range

Table A-9: CML Grammar (continued)

CML TAG/ELEMENT	DESCRIPTION
numeric-range	gt-range / ge-range / lt-range / le-range / eq-range
string range	string-literal / regular-exp
gr-range	int ">"
ge-range	int ">="
lt-range	">" int
le-range	">=" int
eq-range	"=" int
string-literal	<"> string <">
regular-exp	"r" <"> string <">
option	;" option-elem
option-elem	option-name / option-nv
option-nv	option-nv
option-name	string
option-value	string

CML Options

Table A-10 lists several common CML tags you can use in your Application Configuration Templates.

Table A-10: CML Options

NAME	DESCRIPTION
Global Only Options	
filename-key=<key> (no default value) filename-default=<filename> (no default value)	filename-key identifies a path to the key in a valueset that will contain the filename of the file being generated. filename-default identifies the default filename the will be returned if there is no filename in the valueset.
full_template (default) partial_template	full_template is the default behavior and indicates that all expected data in the file must be modeled in the template. partial_template indicates that unmatched data in the file should be ignored and passed directly through to the output. This option only works with preserve-format.
timeout (default value is 0)	timeout represents the number of minutes that should be added onto the Configurations total timeout. A valid timeout is any integer from 0-999 (inclusive). The timeouts of all the templates in a configuration get added together, and that number is added to the default timeout for configurations (as of the time of this update, 10 minutes) to get the final timeout value for the entire configuration.
Regular Options	

Table A-10: CML Options (continued)

NAME	DESCRIPTION
<p><code>unordered-lines</code> (default)</p> <p><code>ordered-lines</code></p>	<p><code>ordered-lines</code> instructs the parser that child tags of the template object (lines, loops, conditionals, etc.) must appear in the file in the ordered they are specified in the template.</p> <p><code>unordered-lines</code> allows child tags of template to appear in any order; however, position of items within ordered sequence elements is preserved.</p> <p>Valid for groups.</p>
<p><code>ordered-elements</code> (default)</p> <p><code>unordered-elements</code></p>	<p><code>ordered-elements</code> instructs the parser that child tags of the group object (loops, conditionals, elements, and so on) must appear in the file in the ordered they are specified in the template.</p> <p><code>unordered-elements</code> allows child tags of of the current group to appear in any order; however, position of items within ordered sequence elements is preserved.</p> <p>Valid for groups.</p>
<p><code>relaxed-whitespace</code> (default)</p> <p><code>strict-whitespace</code></p>	<p><code>strict-whitespace</code> requires that whitespace in the template be matched exactly in the file.</p> <p><code>relaxed-whitespace</code> allows whitespace in the template to be matched by any combination of tabs and spaces.</p>
<p><code>required-whitespace</code> (default)</p> <p><code>optional-whitespace</code></p>	<p><code>required-whitespace</code> requires that whitespace in the template be in the file.</p> <p><code>optional-whitespace</code> makes the presence of non-significant whitespace in the file optional.</p>

Table A-10: CML Options (continued)

NAME	DESCRIPTION
<p>missing-values-are-null (default)</p> <p>missing-values-are-error</p>	<p>missing-values-are-null instructs that values that are not found in the file are null, and therefore not provided in the valueset.</p> <p>missing_values_are_error throws an error if all values specified in a template are not found in a file or valueset.</p>
<p>case-insensitive-keywords (default)</p> <p>case-sensitive-keywords</p>	<p>case-sensitive-keywords instructs that literal text in the template must be matched in a case-sensitive basis in the file.</p> <p>case-insensitive-keywords allows case-insensitive matching of literal text in the file.</p>
<p>required (default)</p> <p>optional</p>	<p>required elements must be matched (unless nested inside optional groups).</p> <p>optional elements are optional.</p> <p>Valid for any tag</p>
<p>skip-lines-without-values (default)</p> <p>show-lines-without-values</p>	<p>skip-lines-without-values instructs when a line has replace elements, and all values for those elements are null, that line should be suppressed from the output.</p> <p>show-lines-without-values instructs that all lines should be shown, regardless of the presence or absence of null values.</p>

Table A-10: CML Options (continued)

NAME	DESCRIPTION
<p>skip-groups-without-values (default)</p> <p>show-groups-without-values</p>	<p>skip-lines-without-values instructs when a group has replace elements, and all values for those elements are null, that groups should be suppressed from the output.</p> <p>show-lines-without-values instructs that all groups should be shown, regardless of the presence or absence of null values.</p>
<p>sequence-append (default)</p> <p>sequence-replace</p> <p>sequence-prepend</p>	<p>sequence-replace indicates that sequence elements child scopes replace sequence elements in parent scopes.</p> <p>sequence-append sequence elements child scopes are appended to sequence elements in parent scopes.</p> <p>sequence-prepend sequence elements child scopes are prepended to sequence elements in parent scopes.</p> <p>Valid for loops and sequences.</p>
<p>not-primary-field (default)</p> <p>primary-field</p>	<p>not-primary-field indicates this field should not be used for the purposes of identifying duplicate items when performing list aggregation.</p> <p>primary-field indicates this field should be used for the purposes of identifying duplicate items when performing list aggregation</p> <p>Valid for sequence and replace tags inside a sequence.</p>

Table A-10: CML Options (continued)

NAME	DESCRIPTION
namespace=<namespace> (default is "/")	namespace identifies the namespace that elements with unqualified names (names without a preceding slash or period) will be stored in.
boolean-yes-format=<string> (default="yes") boolean-no-format=<string> (default="no")	boolean-yes-format and boolean-no-format identifies the strings that will be used to match boolean elements. Valid for boolean replace tags
line-comment=<string> (no default) line-comment-is-whitespace line-comment-is-comma line-comment-is-semicolon line-comment-is-tab	line-comment sets the character that indicates that the remainder of the line will be parsed as a comment.
sequence-delimiter=<string> sequence-delimiter-is-whitespace (default) sequence-delimiter-is-comma sequence-delimiter-is-semicolon sequence-delimiter-is-tab	sequence-delimiter sets the character that separates items within a sequence. Valid for sequences.
field-delimiter=<string> field-delimiter-is-whitespace (default) field-delimiter-is-comma field-delimiter-is-semicolon field-delimiter-is-tab field-delimiter-is-eol	field-delimiter sets a character that will be used to terminate parsing for a replace element value. Valid for replace tags and sequence tags.

Table A-10: CML Options (continued)

NAME	DESCRIPTION
<code>line_continuation=<string></code>	<code>line_continuation</code> sets a character that will be used to indicate that the current line in a config file should be wrapped to the subsequent line.

Index

A

- accessing
 - CDR214
- adding
 - hardware support to Linux build images205
 - NIC support to Windows floppy images201
 - package, new content to261
- address ranges, changing in IP ranges 45
- Application Configuration Management
 - sequence merging
 - append mode303
 - prepend mode305
 - primary key option304
 - replace mode303
- audit
 - creating, packages256

B

- boot floppies
 - Opsware Build Image Administrator options ..204
 - Windows servers
 - creating for203
 - overview199
- build customization scripts
 - Linux, overview181
 - overview171
 - requirements
 - for Linux181
 - for Solaris176
 - Solaris
 - overview176
 - sample177
- build images, adding hardware support for Linux 205

C

- CDR. See code deployment.
- CIDR, changing in IP ranges 45
- CML
 - about the parser271
 - about the tutorial269

- anatomy of a tag271
- Application Configuration Template, defined ..270
- Application Configuration, defined270
- completed templates (url_scan_ini.tpl)297
- creating a template275
- defined271
- fundamentals270
- grammar306
- important tags273
- options309
- sequence aggregation301
- using DTD tags300
- code deployment
 - access control, overview for setting up226
 - accessing214
 - code and content, uploading to staging211
 - configuration
 - checklist218
 - planning221
 - procedures218
 - steps219
 - troubleshooting246
 - creating, directories on hosts225
 - determining, deployment requirements220
 - directories, populating initial content226
 - features213
 - hosts, preparing for CDR225
 - pre- and post-synchronization scripts, details of
 - running235
 - sequences
 - creating242
 - deleting245
 - modifying245
 - services
 - creating230
 - modifying235
 - synchronizations
 - defining237
 - deleting240
 - modifying240
- Code Deployment and Rollback (CDR). See code deployment.

configuration	
checklist for CDR	218
Code Deployment and Rollback feature	219
planning for CDR	221
procedures for CDR	218
troubleshooting for CDR	246
conventions used in the guide	16
creating	
CDR directories	225
CDR sequences, overview	241
CDR synchronizations, overview	237
CML template	275
customer accounts	28
deployment stage values	38
directories on hosts for code deployment	225
IP range groups	43
Linux boot image	206
media resource locators (MRLs)	146
package	
audit results, from	256
managed server, from	255
native formats, in	257
snapshot, from	257
sequences for CDR	242
server use values	36
services for CDR	230
synchronizations for CDR	237
Windows boot floppies	203
custom attributes	
Linux OS provisioning, setting for	192
Solaris OS provisioning, setting for	191
Windows OS provisioning, setting for	193
customer accounts	
creating	28
deleting, customers	34
setting, custom attributes	32
updating	30
customers	
association with servers	26
Customer Independent, definition of	25
Not Assigned customer, definition of	25
D	
defining, synchronizations for CDR	237
deleting	
customers	34
deployment stage values	40
media resource locators (MRLs)	148
OS installation profiles	170
sequences for CDR	245
server use values	38
services for CDR	236
synchronizations for CDR	240
deployment. See code deployment.	
DHCP	
Linux servers, requirements for using	205
OS provisioning, usage of	135
Solaris servers, booting with	173
directories	
code deployment, creating for	225
populating initial content for CDR	226
E	
editing	
deployment stage values	40
media resource locators (MRLs)	147
server use values	37
encoding	
install scripts	266
package metadata	263
examples	
response file	
for Windows 2000	153
for Windows NT	154
sample mapfile for Intel Ethernet Adapter	202
sample Solaris build customization script	177
Windows sample mapfile	202
F	
floppy images, prerequisites for Windows	203
G	
grammar, in CML	306
H	
hardware support	
adding to Linux build images	205
OS provisioning	198
histories	
viewing, changes in OS installation profiles	169
HKEY_CLASSES_ROOT	252
HKEY_CURRENT_CONFIG	252
HKEY_LOCAL_MACHINE	252
HKEY_USERS	252

I

- Import Media Tool, creating MRLs146
- IP addresses
 - changing, CIDR in IP ranges 45
 - prefix length, decreasing in IP ranges 47
 - ranges, changing 45
 - status in IP range, changing 47
- IP range groups
 - creating 43
 - overview 41
- IP ranges
 - changing address ranges for 45
 - CIDR, changing 45
 - decreasing, prefix length 47
 - IP address status, changing 47
 - overview 41

L

- Linux
 - build customization scripts
 - overview181
 - requirements for181
 - creating, boot image206
 - hardware support, adding to build images205
 - PXE, using for booting servers198
 - setting, custom attributes for servers192

M

- mapfiles
 - sample for Intel Ethernet Adapter202
 - sample for Windows servers202
- media resource locators (MRLs)
 - creating146
 - creating, prerequisites for141
 - deleting148
 - editing147
- modifying
 - sequences for CDR245
 - sequences for CDR, overview241
 - services for CDR235
 - services for CDR, overview229
 - synchronizations for CDR240
 - synchronizations for CDR, overview237

N

- NIC support, Windows servers, adding201

O

- operating systems
 - defining for OS provisioning155, 159
- Opware administrators
 - creating, customers28
 - deleting, customers34
 - restrictions for deleting customers33
 - setting, custom attributes32
 - updating, customer information30
- Opware Build Image Administrator, options for .204
- Opware guides
 - contents15
 - conventions used16
 - documentation set17
 - icons in guide, explained17
- Opware SAS
 - Code Deployment & Rollback, features213
 - documentation set17
 - related documentation17
 - server attributes35
 - supported operating systems21, 24
- OS build process
 - default values for190
 - Solaris servers172
- OS installation profiles
 - deleting170
 - histories, viewing169
 - modifying164
 - modifying packages in167
 - overview149
 - properties, changing164
 - software, specifying150
 - working with155
- OS media
 - management, overview140
 - media resource locators (MRLs), creating146
 - prerequisites for creating MRLs141
- OS provisioning
 - hardware support198
 - Linux
 - custom attributes, setting up192
 - modifying operating system installation164
 - OS installation profiles, preparing155, 159
 - Prepare Operating System Wizard155, 159
 - setup
 - overview134
 - Solaris custom attributes, setting up191
 - Windows custom attributes, setting up193

P

package types	
depot	253
LPP	253
MSI	248, 253
RPM	248, 253
Solaris	248, 253
packages	
modifying in OS installation profiles	167
overview	252
packaging server	
setting up	248
setting, preferences	251
packaging, overview.	254
permissions	
code deployment, required for	226
sequence role for CDR, defined	227
service role for CDR, defined	227
special deployment role for CDR, defined	227
synchronization role for CDR, defined	227
preferences	
packaging servers, SAS Client	251
prefix length	
decreasing for IP ranges	47
increasing for IP ranges	46
Prepare Operating System Wizard	155, 159
prerequisites	
MRLs, creating	141
Windows floppy images, creating	203
properties, OS installation profiles, changing for	164
PXE images	
overview for Windows and Linux	198
Windows, modifying for	204

R

Red Hat Linux	249
response files	
example	
for Windows 2000	153
for Windows NT	154
roles. See user roles.	
RPM	266

S

scripts	
Linux build customization scripts	181
Linux servers, customizing build	181
pre- and post-synchronization scripts for CDR,	
running	235

Solaris build customization scripts, requirements for	176
Solaris servers, customizing build	176
sequence aggregation, with CML	301
sequence merging, application configuration	
append mode	303
prepend mode	305
primary key option	304
replace mode	303
sequences	
creating for CDR	242
deleting for CDR	245
modifying for CDR	245
server attributes	
creating, deployment stage values	38
creating, server use values	36
deleting, deployment stage values	40
deleting, server use values	38
editing, deployment stage values	40
editing, server use values	37
overview	35
servers	
association with customers	26
creating a package	255
preparing for code deployment with CDR	225
services	
creating for CDR	230
deleting for CDR	236
modifying for CDR	235
setting	
custom attributes	32
packaging server	248
preferences, packaging servers for	251
setup for servers	
Linux OS provisioning	136
operating systems for provisioning	155, 159
overview for OS provisioning	134
Solaris OS provisioning	135
Windows OS provisioning	138
snapshot	
creating	
package	257
software	
specifying in OS installation profiles	150
Solaris	
build customization scripts	
overview	176
sample	177
custom attributes, setting for Solaris servers	191
requirements for build customization scripts	176
specifying	

- options, new package content for264
- supported operating systems
 - for managed servers21
 - for SAS Client24
- synchronizations
 - creating for CDR237
 - deleting for CDR240
 - modifying for CDR240

T

- troubleshooting, CDR configuration246

U

- updating, customer accounts30
- uploading
 - code and content to staging for CDR211
- user roles
 - sequence role for CDR227
 - service role for CDR227
 - special deployment role for CDR227
 - synchronization role for CDR227
- using DTD tags in CML300

V

- viewing
 - CDR configuration246
 - changes for OS installation profiles169
 - package details267
- Visual Packager
 - adding, new content261
 - creating packages257
 - audit results, from256
 - managed server, from255
 - methods255
 - snapshot, from257
 - overview247
 - package, overview252
 - packaging, overview254
 - setting, packaging server up248
 - specifying, options, new package content for .264
 - viewing, details267

W

- Windows floppy images
 - NIC support, adding201
- Windows servers
 - boot floppies

- creating203
 - overview199
- floppy images, prerequisites for creating203
- PXE images, modifying204
- PXE, using for booting198
- sample mapfile to build servers202
- sample response file
 - for Windows 2000153
 - for Windows NT154
- setting custom attributes for193
- wizards
 - Prepare Operating System155, 159

Z

- ZIP266

