

HP Server Automation

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

Software Version: 7.81

Policy Setter Guide

Document Release Date: November 2009
Software Release Date: November 2009



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 2000-2009 Hewlett-Packard Development Company, L.P.

Trademark Notices

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

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

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

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

UNIX® is a registered trademark of The Open Group.

Documentation Updates

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

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

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

<http://support.openview.hp.com/selfsolve/manuals>

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

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

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

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

Support

Visit the HP Software Support Online web site at:

www.hp.com/go/hpsoftwaresupport

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

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

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

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

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

To find more information about access levels, go to:

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

Contents

1	Server Management Configuration	11
	SA Customer Accounts	11
	Associating Servers with Customers	12
	Customer Account Administration	14
	Creating a Customer	14
	Updating Customer Information and Settings	16
	Setting Custom Attributes for Customers	17
	Restrictions for Deleting Customers	18
	Deleting a Customer	18
	Server Attributes	19
	Creating Server Use Values	20
	Editing Server Use Values	21
	Deleting Server Use Categories	21
	Creating Deployment Stage Values	22
	Editing Deployment Stage Values	23
	Deleting Deployment Stage Values	23
	IP Range Groups and IP Ranges	24
	Overview of IP Range Groups and IP Ranges	24
	Creating an IP Range Group	25
	Creating an IP Range	25
	Changing Address Ranges on IP Ranges	26
	Increasing and Decreasing the Prefix Length	27
	Changing the Status of an IP Address in an IP Range	28
2	Software Management Setup	31
	Overview of Software Management	31
	Policy-Based Software Management	31
	Software Management Features	32
	Software Management Process	33
	Software Management Setup Tasks	34
	Library	34
	Overview of Software Policies	36
	Managing Software Resources Using a Software Policy	36
	Setting Custom Attributes For Software Policies	37
	Including ISM Controls in Software Policies	37
	Software Policies for Patch Installation	38
	Software Policies for Script Execution	39
	Software Policy Management Tasks	39
	Creating Software Policies and Software Templates	39

Opening a Software Policy or Template	42
Editing Software Policy Properties	43
Adding Software Resources to a Software Policy	44
Viewing Properties of Software Resources in a Software Policy	45
Specifying the Installation/Uninstallation Order in a Software Policy	45
Setting Installation and Update Options for an RPM Package	47
Removing a Software Resource from a Software Policy	48
Adding Custom Attributes to a Software Policy	49
Editing Custom Attributes in a Software Policy	49
Deleting Custom Attributes from a Software Policy	49
Adding Custom Attributes to Servers	50
Duplicating Zip Packages	50
Editing the ZIP Installation Directory	51
Viewing Servers Attached to a Software Policy	51
Viewing All the Software Policies Associated with a Software Policy	52
Viewing OS Sequence Associated with a Software Policy	52
Viewing the History of a Software Policy	53
Locating Software Policies in Folders	53
Folders	53
Folders and Permissions	55
Creating a Folder	55
Setting Folder Properties	56
Deleting a Folder	57
Overview of Package Management	58
AIX Packages	60
HP-UX Packages	61
Linux Packages	63
Solaris Packages	64
Windows Packages	65
ZIP Packages	66
Windows Performance for Uploading Packages	67
Character Encoding for Package Metadata and Scripts	68
Importing Packages	69
Importing Application Installation Media	70
Importing Executables	71
Exporting a Package	72
Ways to Open a Package	72
Viewing Package Properties	73
Editing Package Properties	76
Viewing Package Contents	78
Viewing Servers Associated with a Package	78
Viewing All Software Policies Associated with a Package	78
Deleting a Package	79
Renaming a Package	79
Locating Packages in Folders	79
RPM Deployment	80
Automatically Updating RPM Packages in a Software Policy	81

Upgrade Option for an RPM Package	82
Uninstalling RPM Packages.	84
Software Policy Compliance for RPM Packages	85
Automatically Importing Red Hat Network Errata	85
Viewing Errata Based and Channel Based Software Policies in the SA Client	86
3 Operating System Provisioning Setup	89
OS Provisioning Setup	89
Overview	89
OS Provisioning Setup Tasks	90
Setting Up Sun Solaris OS Provisioning	90
Setting Up Linux or VMware ESX OS Provisioning	91
Setting Up Microsoft Windows OS Provisioning	92
OS Media Management	93
Overview	93
Setting Up the Media Server	94
Import Media Tool Syntax and Options.	99
Creating an MRL with the Import Media Tool	101
Editing an MRL	101
Deleting an MRL.	102
OS Installation Profiles	103
Overview	103
Specifying Software in OS Installation Profiles	104
Configuration Files	104
Sun Solaris Profiles	105
Red Hat Linux Configuration Files	105
VMware ESX Configuration Files	105
SUSE Linux Configuration Files	106
Microsoft Windows Response Files	106
Using OS Installation Profiles	108
Defining an OS Installation Profile — Unix	108
Hardware Signature Files for Windows	111
Defining an OS Installation Profile — Windows	112
Modifying Existing OS Installation Profiles	116
Changing the OS Installation Profile Properties	116
Modifying How an OS Is Installed on a Server — Unix	117
Modifying How an OS Is Installed on a Server — Windows	117
Modifying the OS Installation Profile Packages	119
Viewing Change History for an OS Installation Profile	120
Deleting an OS Installation Profile	121
Build Customization Scripts	122
Using Build Customization Scripts	122
Before Creating a Sun Solaris Build Customization Script	123
Solaris Provisioning from a Boot Server on a Red Hat/SLES 10 Linux Server	125
Solaris Build Customization Script	125
Requirements for Solaris Build Customization Scripts	126
Sample Solaris Build Customization Script	127

Linux Build Process	127
Linux Build Customization Scripts	129
Requirements for Linux Build Customization Scripts	129
VMware ESX Build Process	130
VMware ESX Build Customization Scripts	130
Windows Build Process (DOS Boot Image)	130
Windows Build Process (WinPE Boot Image)	131
Windows Build Customization Scripts (DOS Boot Image)	133
Default Custom Attribute Values	135
Custom Attributes for Sun Solaris	136
Custom Attributes for Linux or VMware ESX	137
Custom Attributes for Microsoft Windows	139
Adding Custom Attributes to OS Installation Profile (SAS Web Client)	141
Adding Custom Attributes to OS Installation Profile (SA Client)	142
Virtualization Support — VMware ESX (and Solaris 10)	142
Hardware Support in OS Provisioning	142
Overview of Hardware Support in OS Provisioning	143
PXE Images for Windows and Linux or VMware ESX	143
Windows and Linux or VMware ESX Boot Images	143
NIC Support in Solaris Boot Images	144
NIC Support in Red Hat Linux Boot Images	144
NIC Support in Windows Boot Images	144
Adding NIC Support to a Windows Boot Image	146
Sample Mapfile	147
Sample Mapfile for an Intel 8255x-based PCI Ethernet Adapter	147
Prerequisites for Creating Windows Boot Images	148
Creating a Windows Boot Image	148
Updating PXE Image for Windows	149
Adding Hardware Support to a Linux or VMware ESX Build Image	150
Creating a Linux or VMware ESX Boot Image	150
Example: Usage of the OPSWlinuxbootiso Utility	151
4 Software Discovery	153
Software Discovery Prerequisites	153
Generating Reports with SAR	154
Supported Platforms and Configurations	154
Downloading the Software Discovery Package	154
Contents of the Software Discovery Package	155
Deploying Software Discovery Using Inventory Snapshots	156
Viewing Discovered Software on a Managed Server	156
Inventory Snapshot Job Error Messages	157
Software Discovery Permissions	158
How Software Discovery Works	158
Application Discovery	158
Application Signatures	159
Configuration and Customization	164
Configuration Attributes	164

Configuration Options	166
Syntax Errors	167
Discovery Filtering Process	167
How Filters Affect Scanning	167
Sample Configuration	167
Concurrency and Multi-User Considerations	168
Software Discovery Usage Examples	169
Example 1	169
Example 2	169
Example 3	170
Extending Software Discovery	170
Writing Custom Software Discovery Code	171
5 Code Deployment Setup	173
Code Deployment Process	173
Deploying Code	173
Uploading Code and Content to Staging	175
CDR Operations and Directories	175
CDR Features	176
CDR Permissions	176
Accessing CDR	177
Code Deployment & Rollback Setup	179
Prerequisites for Code Deployment & Rollback	179
Code Deployment Configuration Checklist	180
Planning and Defining a CDR Configuration	180
Configuring a Site to Use CDR for Code and Content Updates	181
Code and Content Deployment Requirements	182
Overview of CDR Configuration Planning	182
Preparing Host Machines	186
Creating or Verifying Directories on Hosts	186
Initial Content in Directories	186
Access Control for CDR	187
Defining CDR Services, Synchronizations, and Sequences	189
CDR Service Management	189
Defining a Service	189
Pre-Synchronization and Post-Synchronization Scripts	193
Modifying a Service	193
Deleting a Service	194
CDR Synchronization Management	194
Defining a Synchronization	195
Modifying a Synchronization	196
Deleting a Synchronization	197
CDR Sequence Management	197
Defining a Sequence	198
Modifying a Sequence	200
Deleting Sequences	201
Verifying and Troubleshooting CDR Configuration	201

Index 203

1 Server Management Configuration

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

- Creating customers in SA 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 SA users install the Agent on servers running in the operational environment
- Creating and modifying the values for Server Attributes so that SA 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.

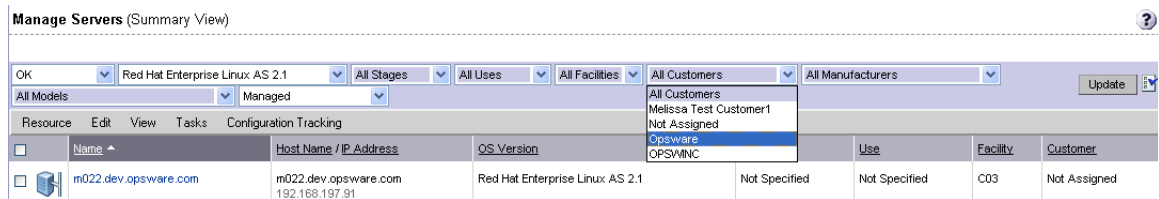
SA Customer Accounts

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).

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

When an SA administrator creates a customer in SA, a value for that customer is automatically added to the customer filter in the Managed Servers list, as [Figure 1](#) shows.

Figure 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, SA is shipped with the following two customers:

- **Customer Independent:** A global customer in SA. 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 SA, the server is associated with the Not Assigned customer if IP ranges were not created to automatically associate assimilated servers with customers. See [Figure 2](#).



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

Figure 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

Associating Servers with Customers



The information in this section applies only to servers whose Agents communicate directly with an SA Core. Servers with Agents that use an SA Gateway to communicate with the Core must be manually associated with the customer after the Agent is installed on the server. For more information about manually associating a Managed Server with a customer, see the *User's Guide: Server Automation*

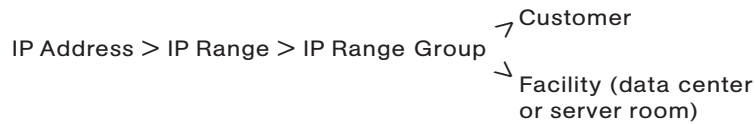
An SA user or an SA administrator can set up an IP range group so that servers are automatically associated with customers when users perform the following server management tasks:

- Manage servers running in the operational environment by installing an Agent on the servers
See the *User's Guide: Server Automation* for more information on how to install an SA Server Agent.
- Use the OS Provisioning feature to install operating systems on bare-metal servers
See the *SA 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 24 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.

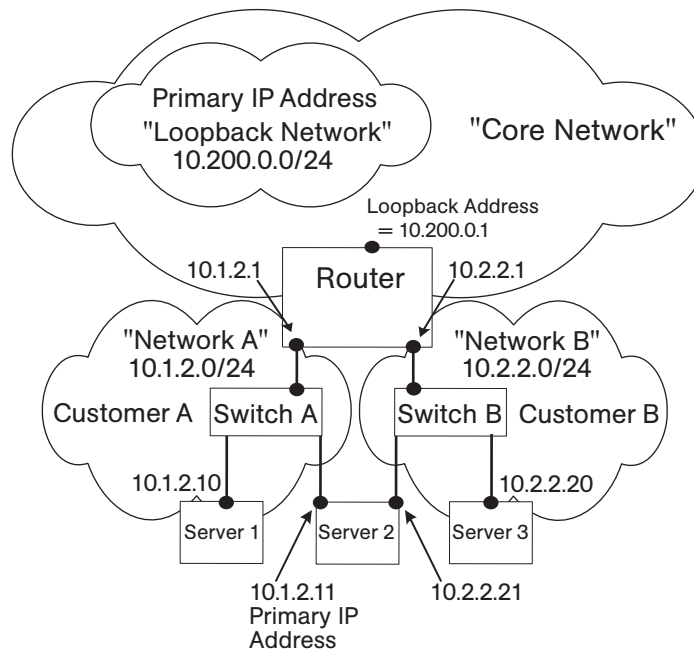
See *User's Guide: Server Automation* for information about how to change the customer association for a server.

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

See the *User's Guide: Server Automation* for more information about how SA 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. SA enforces the relationship between server and facility at hardware registration. See [Figure 3](#).

Figure 3 Primary IP Addresses in SA



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

As an SA administrator, you can add customers, update information and configuration settings for an existing customer, and delete customers. This section provides information about customer account administration within SA and contains the following topics:

- [Creating a Customer](#)
- [Updating Customer Information and Settings](#)
- [Setting Custom Attributes for Customers](#)
- [Restrictions for Deleting Customers](#)
- [Deleting a Customer](#)

Creating a Customer



As an SA administrator, you can add customers to your SA installation to create designations by business unit to provide management of SA 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 4](#) shows.

Figure 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 5](#) shows.

Figure 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"/> <small>Must consist of uppercase letters, numbers, '-', and '_'.</small>
	<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 SA 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 17 for more information.

Updating Customer Information and Settings

As an SA administrator, you can update information or change configuration settings for existing customer accounts.

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 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 SA-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.

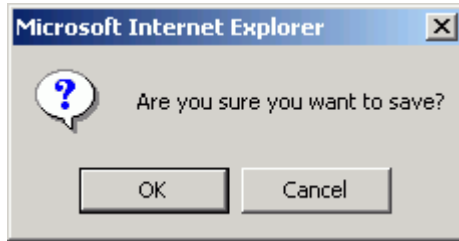
Figure 6 Assign Facility Section of the Customers: Edit Properties Page

The screenshot shows the 'Customers: Edit Properties' page. 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 fields: ID (1220007), Name (Important Customer), Short Name (VIP), and Status (ACTIVE). Below the table is the 'Assign Facility' section, which consists of two list boxes: 'Assigned Facility' (empty) and 'Available Facility' (containing 'C07'). Between the list boxes are two arrows: a right-pointing arrow (→) and a left-pointing arrow (←). At the bottom of the page 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 7](#) shows.

Figure 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 8](#) shows.

Figure 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 SA, 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 SA Support Representative to help you determine the appropriate changes to make when you update the information or settings for a specific customer.

- 5 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 SA 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 18 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 9](#) shows.

Figure 9 Customers: Delete Confirmation Page

[Return to Customers](#)

The following selected customer(s) will be removed.	
	Customer
<input checked="" type="checkbox"/>	sophie Test1

Move deactivated servers to the Not Assigned customer

- 4 Click **Delete**.

A confirmation page appears, as [Figure 10](#) shows.

Figure 10 Delete Customer: Confirmation Page

Delete Customer: Confirmation

Selected Customer(s) has been removed.

- 5 Click **Continue**.

The SAS Web Client displays the updated Customers page.

Server Attributes

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

- [Associating 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.

SA 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 *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 11](#) shows.

Figure 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 12](#) shows.

Figure 12 Create Server Use Value Page

Create Server Use Value

[Return to Server Use](#)

Name:	<input type="text"/>
Description:	<input type="text"/>
Code Deployment:	<input type="checkbox"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

- 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 13](#) shows.

Figure 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:

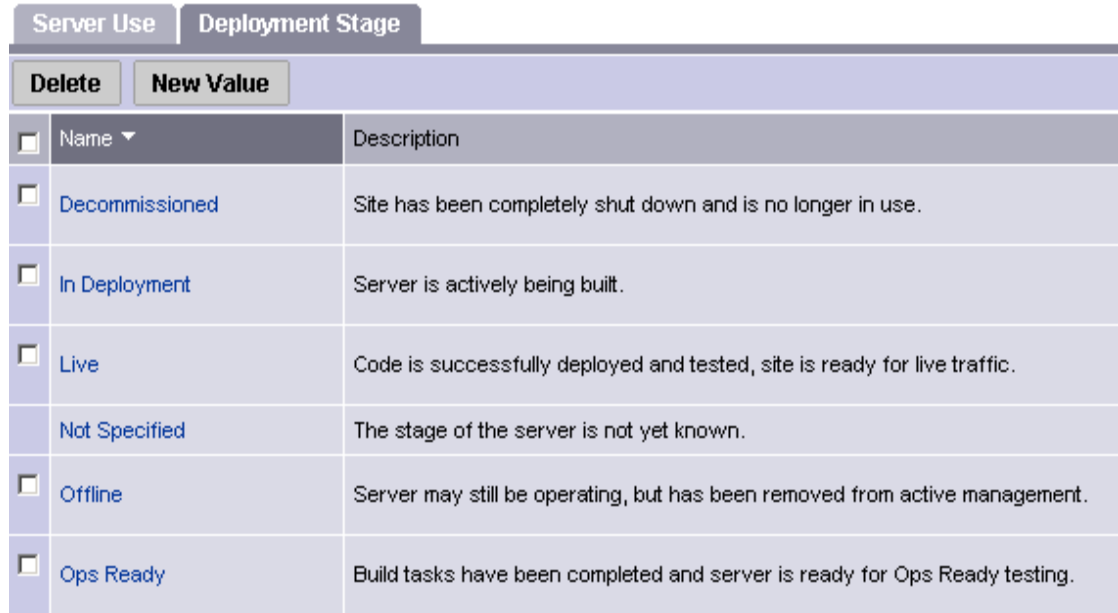
- 1 From the navigation panel, select Administration ► Server Attributes. The Server Attributes page appears with the Server Use tab displayed, which shows all previously-defined server use categories.
- 2 Select the check box next to each of the server use categories that you want to delete. You cannot 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 14](#) shows.

Figure 14 Deployment Stage Tab of the Server Attributes Function



<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.
	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 15](#) shows.

Figure 15 Create Deployment Stage Value Page



Create Deployment Stage Value

[Return to Deployment Stage](#)

Name:

Description:

Save **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 16](#) shows.

Figure 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 SA 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 SA user or an SA 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 Agent on the servers

See the *User's Guide: Server Automation* for more information on installing an 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 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 14 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 17](#) shows.

Figure 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 18 shows.

Figure 18 IP Range Groups Page in the SAS Web Client

<input type="checkbox"/>	Name	Customer
<input type="checkbox"/>	CORP	Opsware
<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 19 shows.

Figure 19 IP Ranges for the Default Customer (“Not Assigned”)

<input type="checkbox"/>	IP Range Name	Pool Name	Description	IP Range Type	Sub-Type	Subnet / CIDR
<input type="checkbox"/>	Default	Default	Holding pool for IPs used by Devices but not managed as part of other VLANs	PUBLIC	PRODUCTION	n/a/-1

- 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.
- Click **New** at the top of the page. The IP Ranges: Create IP Range Type page appears, as [Figure 20](#) shows. You can add up to five new IP ranges at a time.

Figure 20 Creating an IP Range

IP Ranges: Create IP Range Type Customer UNKNOWN Facility"CD7"			
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"/>		

- 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.
- 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 SA provides a way of specifying a range of IP addresses to include in an IP range.

SA 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:

- From the navigation panel, click Environment ► IP Ranges. The IP Ranges: View IP Ranges page appears.
- 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 21](#) shows.

Figure 21 IP Ranges in the SAS Web Client

<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.

- 4 To change the current CIDR setting, select a new value from the list in the New CIDR column.
- 5 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 SA.

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 22](#).

Figure 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"/>
	ASSIGNED
	AVAILABLE
	NOT-AVAILABLE
	RESERVED
	NETWORK
	DHCP
	BROADCAST
	GATEWAY
	VIRTUAL
	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**.

2 Software Management Setup

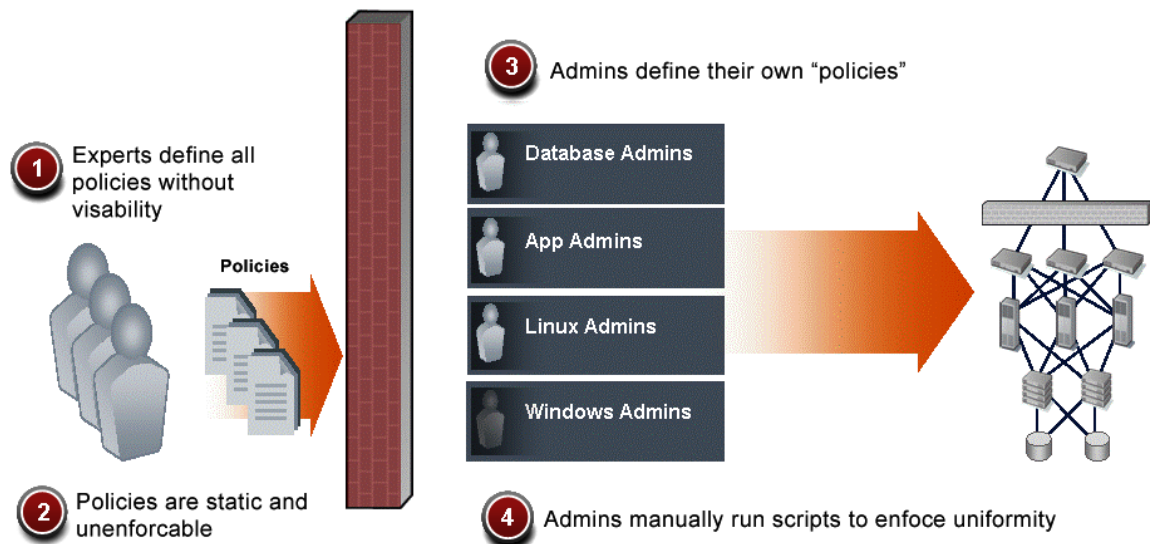
Overview of Software Management

The Software Management feature in SA 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 23](#).

Figure 23 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. SA solves the software deployment problem by introducing the Software Management feature in the SA Client.

Using the Software Management feature allows IT managers to enforce policies, standardize operations, and ensure compliance against defined software policies. In the SA 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 SA, policy setters create a software policy that specifies the software 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. SA 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 SA 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, scripts, software policies, OS sequences, and server objects.

- **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**

SA 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, patches, scripts, and server objects 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. SA 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, scripts, application configurations, and server objects 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.

- **Deploy application installation media on managed servers**

The Software Management feature allows users to import a software application provided by a software vendor into SA and deploy that software application on a managed server.

- **Establish a software Installation process**

The Software Management feature allows you to separate the different stages (analysis, download, and installation) 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.

- **Install and uninstall software directly on managed servers**

The Software Management feature allows you to install and uninstall software directly on a managed server without using a software policy.

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

The Compliance View (for individual servers or groups of servers) 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 SA 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 SA administrator assigns Folder permissions, Client feature permissions, and Customer constraints to define the security boundaries across various user groups. By assigning folder permissions, the SA administrator can specify which user groups can view, use, and modify the software resources in a folder. And by assigning SA Client feature permissions, the SA administrator can specify the actions the users in a user group can perform with the SA Client.

See the *SA 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 SA, creating scripts, 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 34 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 *SA 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 SA, and managing packages in SA. See [Overview of Package Management](#) on page 58 for more information.
- Patch management tasks such as importing patches to SA, and managing patches and patch policies. See the *SA User's Guide: Application Automation* for more information about Patch Management.
- Script Management tasks such as importing and creating scripts in SA and managing scripts. See the *SA User's Guide: Application Automation* for more information about Script Management.
- Application Configuration management tasks such as creating application configurations. See the *SA User's Guide: Application Automation* for more information about Application Configurations.
- 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 53 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, ordering policy items, adding custom attributes and ISM Controls to a software policy, and creating multiple instances of Zip packages. See [Overview of Software Policies](#) on page 36 for more information.
- Creating and managing server objects such as Windows COM+, Windows Registry, User's and Groups, and adding server objects to the Library. See the *SA User's Guide: Application Automation* for more information about Audit and Remediation.

Library

In the SA Client, the Library provides a way to display the software resources (such as application configurations, software policies, patches, patch policies, packages, OS sequences, OS profiles, Windows COM+, Users and Groups, Local Security Settings) managed by SA. Folders are also located in the Library. In the Library, you can view the software resources either by their type or by their location in the folder hierarchy. Some of the software resources

such as patch policies, and application configurations, are not contained in folders, and they can be only viewed in the By Type view. The following table lists the software resources displayed in the Library and whether they can be added to folders.

Table 1 Feature Objects Displayed in the Library in the SA Client

Software Resources	By Type View	By Folder View
Application Configurations	X	NA
Software Policies	X	X
Audit and Remediation	X	X
Patches	X	NA
Patch Policies	X	NA
OS Installation Profiles	X	NA
OS Sequences	X	X
Packages	X	X
Scripts	X	X
Windows Users and Groups	X	X
Unix Users and Groups	X	X
Windows COM +	X	X
Windows Registry	X	X
Windows Services	X	X
Windows IIS Metabase	X	X
.Net Framework Configuration	X	X
Local Security Settings	X	X
Web Applications	X	NA
Databases	X	NA

See the *SA User's Guide: Application Automation* for more information about Audit and Remediation, Application Configuration, OS Provisioning, Script Execution, and Patch Management. See the [Audit and Remediation](#) on page 3 for more information about server objects.

When you install SA, the Library contains the following default folders:

- A folder that contains the tools required to *upload ISMs* to SA.
- The *Package Repository folder* contains packages that are organized by operating system families. See [Overview of Package Management](#) on page 58 for more information.
- The *Home folder* contains a folder for every SA user and only the specified SA user has access to the home folder.

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 55 for more information.

Overview of Software Policies

In SA, software policies allow you to install software and configure applications simultaneously. A software policy can contain packages, RPM packages, patches, application configurations, scripts, and server objects. It can also be associated with OS Sequences. 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, scripts, server objects, 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. SA then determines what operations are required to make the server conform to its model. See the *SA 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 *SA 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 *SA User's Guide: Application Automation* for more information about installing software using a software template.

A software policy can be associated with either a single operating system family or multiple operating system families. 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 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.

Managing Software Resources Using a Software Policy

A software policy can contain packages, RPM packages, patches, application configurations, scripts, and server objects (Windows COM+, Windows Services, windows Registry, Unix Users and Groups, Windows Users and Groups, Local Security Settings, .Net Framework Configurations). See the *SA User's Guide: Server Automation* for more information about server objects.

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, SA installs the software resources in the software policy in 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. Thus, policy inclusion provides a way to organize your software and manage dependencies between the software resources across included policies.

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

See [Specifying the Installation/Uninstallation Order in a Software Policy](#) on page 45 for more information.

- ▶ If a Software Template contains sub policies then during remediation, SA 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 SA 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 SA 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 49 for more information.

Including ISM Controls in Software Policies

An Intelligent Software Module (ISM) is an installable software package created with the 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 SA you can create a control script with a text editor, package the script into an ISM, and then upload the ISM to SA. See the *SA Content Utilities Guide* for more information about ISM control scripts. After you upload an ISM package into SA by using the ISM tool in the

IDK, the ISM appears in the SA Client as a package in the Library. Using the SA Client, you add the ISM package to a software policy and then attach the software policy to managed servers. See [Adding Software Resources to a Software Policy](#) on page 44 for more information.

You can run the control scripts in the ISM with the Run ISM Control window of the SA Client. See the *SA 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 SA 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 SA objects. If a server is associated with objects that have identically named custom attributes, SA uses a predefined search order to determine the custom attribute that provides the parameter value. In the Run ISM Control window of the SA Client, you can view the name and value of the control parameter.

See the *SA Content Utilities Guide* for more information on the search order for custom attributes.

Software Policies for Patch Installation

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

- Using Windows patch policies to install Windows patches. See the [Patch Management for Windows](#) on page 161 for more information.
- Using Solaris patch policies to install Solaris patches. See [Patch Management for Solaris](#) on page 249 for more information.
- Using software policies to install Unix patches. See [Patch Management for Unix](#) on page 221 for more information.
- Installing patches directly on servers. See the *SA User's Guide: Application Automation* for more information about installing patches directly.



A software policy can contain both Unix patches and Windows patches. It is recommended that you use Windows patch policies to install Windows patches, Solaris patch policies to install Solaris patches and software policies to install other Unix patches on servers.

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 *SA 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 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 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.

Software Policies for Script Execution

In the SA Client, you can execute scripts in the following ways:

- Execute a server script directly on servers or server groups. See the *SA User's Guide: Application Automation* for more information about script execution.
- Add a script to a software policy and execute the script by attaching the software policy to the server and then remediating the server against the software policy. See the *SA User's Guide: Application Automation* for more information about installing software.

A software policy allows you to execute multiple scripts on servers or server groups simultaneously, and execute a sequence of scripts on a server by specifying an install order in the software policy.

Software Policy Management Tasks

The software policy management tasks include:

- Creating Software Policies and Software Templates
- Opening a Software Policy or Template
- Editing Software Policy Properties
- Adding Software Resources to a Software Policy
- Specifying the Installation/Uninstallation Order in a Software Policy
- Setting Installation and Update Options for an RPM Package
- Removing a Software Resource from a Software Policy
- Adding Custom Attributes to a Software Policy
- Editing Custom Attributes in a Software Policy
- Deleting Custom Attributes from a Software Policy
- Adding Custom Attributes to Servers
- Duplicating Zip Packages
- Editing the ZIP Installation Directory
- Viewing Servers Attached to a Software Policy
- Viewing All the Software Policies Associated with a Software Policy
- Viewing the History of a Software Policy
- Viewing OS Sequence Associated with a Software Policy
- Locating Software Policies in Folders

Creating Software Policies and Software Templates

A software policy contains software resources such as packages, patches, RPM packages, scripts, application configurations, and server objects that need to be installed on managed servers. A software template is a software policy that can only contain other software policies.

In the SA Client, you can create a software policy or template in the following ways:

- Creating a Software Policy or Template from the By Type View in the Library
- Creating a Software Policy or Template from the By Folder view in the Library



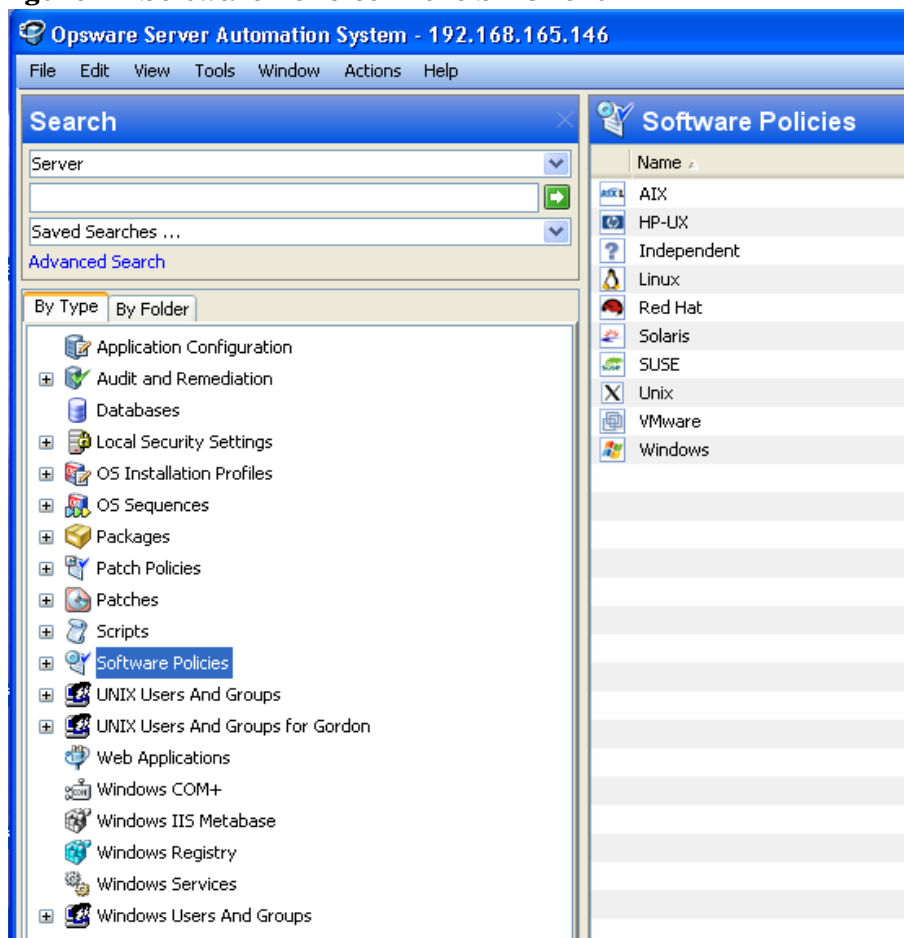
You must have a specific set of permissions to be able to create and manage software policies. To obtain these permissions, contact your SA administrator. See the *SA Administration Guide* for more information.

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

Perform the following steps to create a software policy in the SA 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 24](#). By default, the software policies are organized by operating system families.



Figure 24 Software Policies in the SA Client



- 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 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 Folder 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 SA server life cycle values for the software policy.
- 8 From the OS 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 *SA 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 SA Client, a software policy is represented by the icon . A software template is represented by the icon .

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

Perform the following steps to create a software policy in the SA 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 In the Description field, enter text that describes the purpose or contents of the policy.
- 6 Click **Select** to change the location for the software policy in the folder hierarchy. The Select Folder 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 SA server life cycle values for the software policy.
- 8 From the OS 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 *SA 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 SA Client, a software policy is represented by the icon . A software template is represented by the icon .

Opening a Software Policy or Template

In the SA Client, there are several ways to open a software policy or template. 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

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** ► **Servers** ► **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 **Management Policies** ► **Software Policies**. 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.

Editing Software Policy Properties

After you create a software policy, you can view and modify its properties. You can view properties such as the SA user who created the software policy, the date when it was created, and the SA 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.

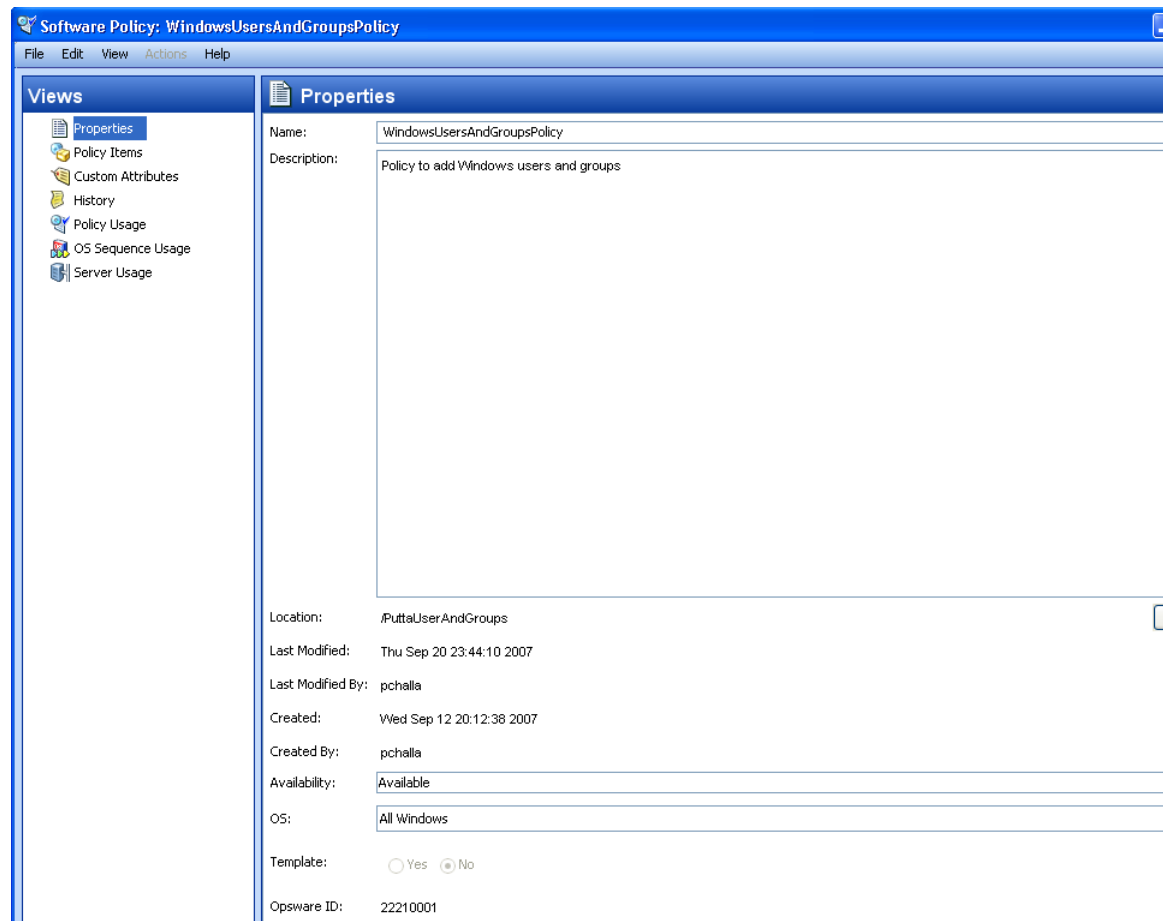


You must have a set of permissions to manage software policies. To obtain these permissions, contact your SA administrator. See the *SA Administration Guide* for more information.

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 25](#).

Figure 25 Software Policy Window



- 3 From the Views pane, select Properties. You can edit the name, description, location, life cycle, and operating systems for the software policy in the Content pane.
- 4 In the Name field, edit the name for the software policy.

- 5 In the Description field, edit the text that describes the purpose or contents of the policy.
- 6 Click **Select** to change the location for the software policy in the folder hierarchy. The Select Folder 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 SA server life cycle values for the software policy.
- 8 From the OS drop-down list, select the operating system family or specific operating systems in that family. If you change the operating system family of an existing software policy, you will see a dialog that shows any policy items that are no longer applicable and/or any servers that are no longer applicable. You should remove non-applicable policy items and detach and remediate any servers that are no longer applicable to the modified policy.
- 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 *SA 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 SA Client, a software policy is represented by the icon . A software template is represented by the icon .

Adding Software Resources to a Software Policy

After you create a software policy, you can add software resources such as patches, packages, application configurations, scripts, and server objects to it. When you add software resources to a software policy, the software resources must contain at least one operating system as that of the software policy. Adding software resources to a software policy does not install them on a managed server. After you add software resources to a software policy, you can install the software directly on the managed server or attach it to a managed server and then remediate the software policy. See the *SA User's Guide: Application Automation* for more information about installing software.

▶ You must have a set of permissions to add software resources to a software policy. To obtain these permissions, contact your SA administrator. See the *SA Administration Guide* for more information.

Perform the following steps to add software resources 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.
- 4 From the **Actions** menu, select **Add Policy Items**. The Select Library Item window appears as shown
- 5 Select Browse Types to display a list of policy items that can be added to the software policy. Select the policy item and click **Select**. The selected policy item appear in the Content pane.

or

Select **Browse Folders** to display the folder hierarchy in the Library and the list of software resources contained in the folders. Select the policy item and click **Select**. The selected policy item appear in the Content pane.

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

Viewing Properties of Software Resources in a Software Policy

Once you have added software resources to a software policy you can view the properties of the software resource in the software policy. For example for a package you can view the install flags and information on the reboot option. For Windows Services you can view the list of services and for Windows patch you can view the install path, install flags, and the information on the reboot information.

You cannot edit the properties of the software resources in the software policy. To edit the properties for a software resource, you must open the specific software resource window and edit.

Only for RPM packages and scripts, you can edit some of the properties in a software policy. For scripts you can specify the command options in a software policy. For RPM packages you can set the installation and update options in a software policy. See [Setting Installation and Update Options for an RPM Package](#) on page 47 for more information.

Perform the following steps to view the properties of a software resource 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. The policy items appear in the content pane.
- 4 Select a policy item. The properties of that policy item appear in the details pane.

Specifying the Installation/Uninstallation 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, scripts, application configurations, included software policies, and server objects in the software policy. When you specify the installation order for the included policies, all the software resources in the included policy are grouped together and installed as a unit.

When you specify the installation order for installing the software resources in a software policy, the software resources are installed in the same order. During Uninstallation, by default, SA will uninstall the software resources (except scripts and application configurations) defined in the software policy in the reverse order.

You can also specify a separate uninstallation order for the software resources in a software policy. The uninstallation order can be completely different from the installation order. During Uninstallation, SA will uninstall all the software resources (except application configurations) defined in the software policy.



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 SA administrator. See the SA Administration Guide for more information.

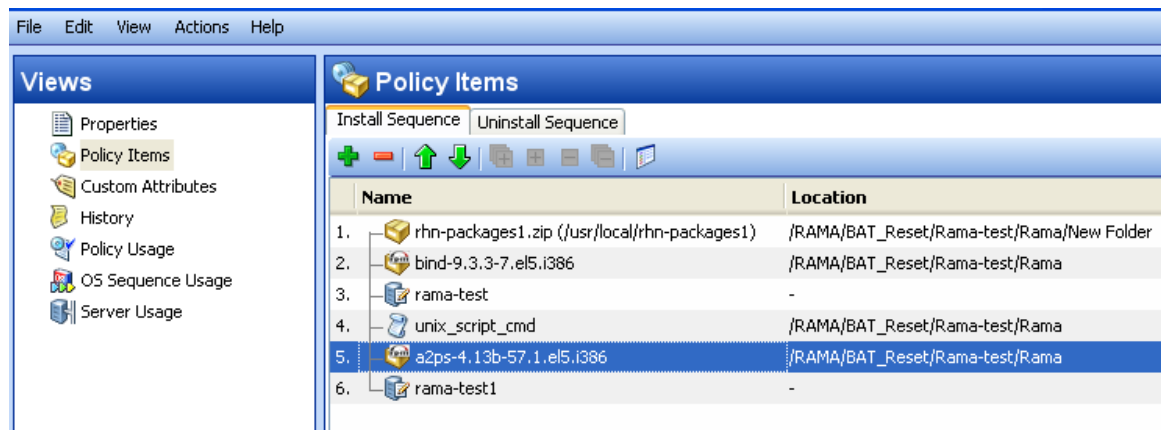
Perform the following steps to specify the installation or uninstallation order in a software policy:

- 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. The list of all the software resources in the software policy appear in the Content pane.
- 4 (Optional) From the Actions menu, unselect **Automatic Uninstall Ordering** to specify the uninstallation sequence.
- 5 Select **Install Sequence** to specify the installation order as shown in [Figure 26](#).

or



Select **Uninstall Sequence** to specify the uninstallation order.

Figure 26 Setting the Installation Order in a Software Policy



- 6 Select the Policy Item and then from the **Actions** menu, select **Move up** or **Move down** to order the policy items.

or

Select the Policy Item and then select  or .

- 7 To save the changes, select **Save** from the **File** menu.
You will be prompted to continue. Click **OK** to continue.

Deleting Install/Uninstall Sequences

You can delete install and uninstall sequences by highlighting the sequence and selecting delete. You will be prompted to confirm the deletion. Click **OK** to continue.

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 attached to the 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 SA administrator. See the *SA Administration Guide* for more information.

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

- 1 From the Navigation pane, select **Library** ► **By Type** ► **Software Policies**.
- 2 From the Content pane, select the software policy containing a RPM package and open it. The Software Policy window appears.
- 3 From the Views pane, select Policy Items. The list of software resources contained in the software policy appears in the Content pane as shown in the figure below.

Figure 27 Setting Installation and Upgrade Options for RPM Packages

The screenshot shows the 'Policy Items' window with two tabs: 'Install Sequence' and 'Uninstall Sequence'. The 'Install Sequence' tab is active, displaying a list of software resources. The list has two columns: 'Name' and 'Location'. The resources are:

Name	Location
1. rhn-packages1.zip (/usr/local/rhn-packages1)	/RAMA/BAT_Reset/Rama-test/Rama/New Folder
2. bind-9.3.3-7.e15.i386	/RAMA/BAT_Reset/Rama-test/Rama
3. rama-test	-
4. unix_script_cmd	/RAMA/BAT_Reset/Rama-test/Rama
5. a2ps-4.13b-57.1.e15.i386	/RAMA/BAT_Reset/Rama-test/Rama
6. rama-test1	-

The 'a2ps-4.13b-57.1.e15.i386' item is selected. Below the list, the configuration for this RPM package is shown:

RPM : a2ps-4.13b-57.1.e15.i386

Install Flags: -

Reboot After Install: No

Install Mode: Upgrade (rpm --upgrade)

Install Criteria: Install RPM always
 Install RPM only if an earlier version is installed

Auto-Update Policy Based On: None

- 4 Select an RPM package. For every RPM package, you can specify the following options:
 - In the Install Criteria, select the option **Install RPM always** to install the RPM packages specified in the software policy on the managed server.
 - In the Install Criteria, select the option **Install RPM only if an earlier version is installed** to update the RPM version on the managed server to the version specified in the software policy.
 - From the Auto-Update based on drop-down list, select **Version or Release** to automatically update the RPM package in the software policy to a newer version or release of the RPM.

▶ SA 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.

- From the Auto-Update based on drop-down list, select **Release Only** to automatically update the RPM package in the software policy to a newer release of the same version of the RPM.

▶ SA 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.

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

Removing a Software Resource from a Software Policy

Removing a software resource from a software policy does not uninstall it from a managed server. It only removes the software resource from the software policy. To uninstall the software resource from a managed server, you must uninstall or remediate the software policy. See the *SA User's Guide: Application Automation* for more information about the uninstalling software.

▶ You must have a set of permissions to remove packages from a software policy. To obtain these permissions, contact your SA administrator. See the *SA 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.
- 4 Select the items that you want to remove from the list of policy items displayed in the Content pane.
- 5 From the **Actions** menu, select **Remove Policy Item**.
- 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 SA administrator. See the *SA 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 SA 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 ► Servers ► 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.

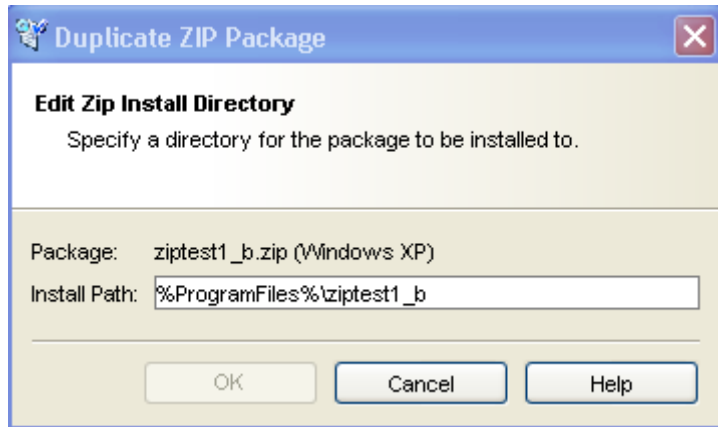
Duplicating Zip Packages

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 SA, 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. SA 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.
- 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 28 Duplicate ZIP Package Window in the SA 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

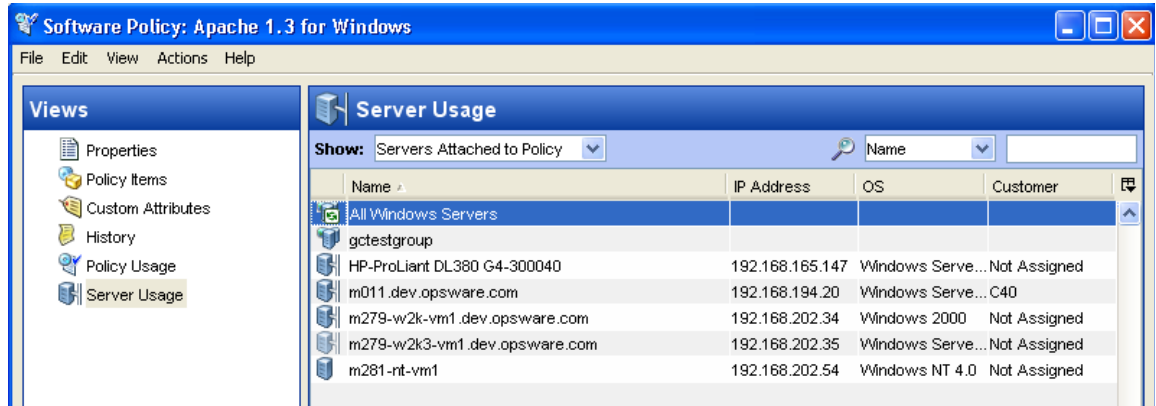
Perform the followings steps to change the default installation directory for ZIP packages:

- 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.
- 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 SA 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 29](#).

Figure 29 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 OS Sequence Associated with a Software Policy

A software policy can be associated with an OS Sequence. In the Software Policy window, you can view all the OS Sequences a software policy is associated with.

Perform the following steps to view the OS Sequence a software policy is associated with:

- 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 OS Sequence Usage. The list of OS Sequences the selected software policy is associated with 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:

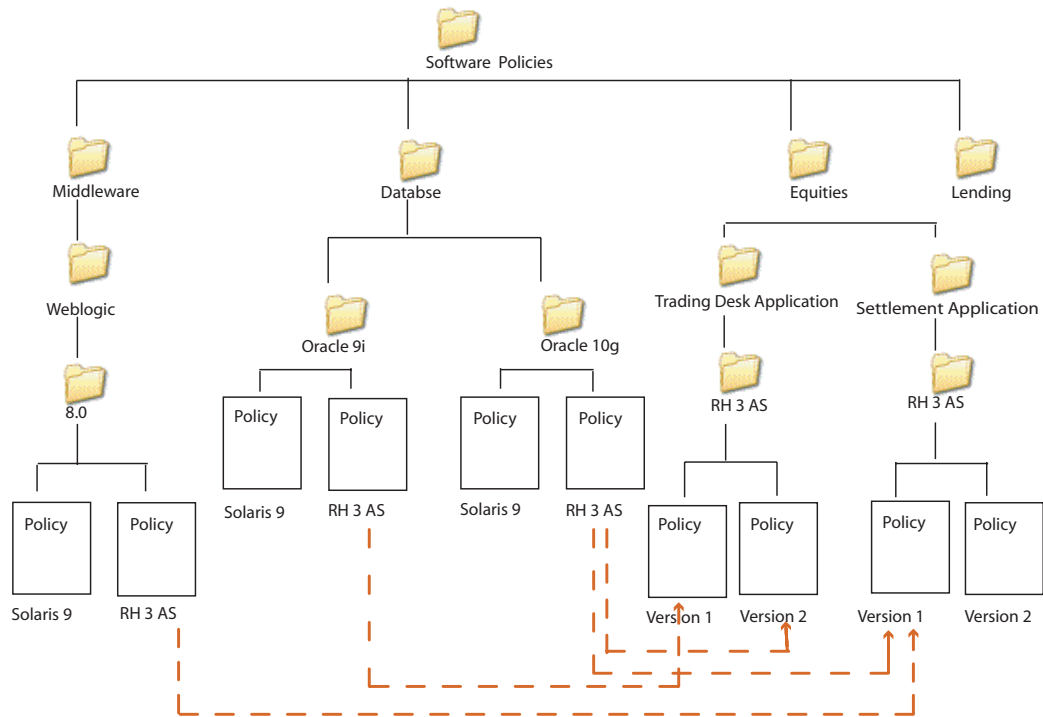
- 1 From the Navigation pane, select **Library ► By Type ► Software Policies**.
- 2 From the Content pane, select the software policy and then select **Locate in Folders** from the **Actions** menu. The folder hierarchy for the software policy appears in the Content pane.

Folders

The folder hierarchy in the SA Client 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, patches, scripts, software policies, server objects, 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 30 Example of a Hierarchical Folder Structure in the SA Client



SA 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 55 and [Overview of Software Policies](#) on page 36 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, patches, server objects, 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 SA Client Feature permissions to access the contents of a folder. SA Client Feature permissions determine what actions users can perform with the SA Client, whereas the Folder permissions specify which folders users have access to.

See the *SA 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.
- **Execute Objects Within Folder:** Enables a user to execute the scripts contained in the folder and view the names of the folder's children. This permission allows users to run scripts, but not to read or write them.

See the *SA Administration Guide* for more information about types of Folder permissions.

In addition to Folder permissions, you can assign customer constraints to folders. See the *SA 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 SA administrator. See the *SA Administration Guide* for more information.

Perform the following steps to create a folder in the SA 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 SA 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.

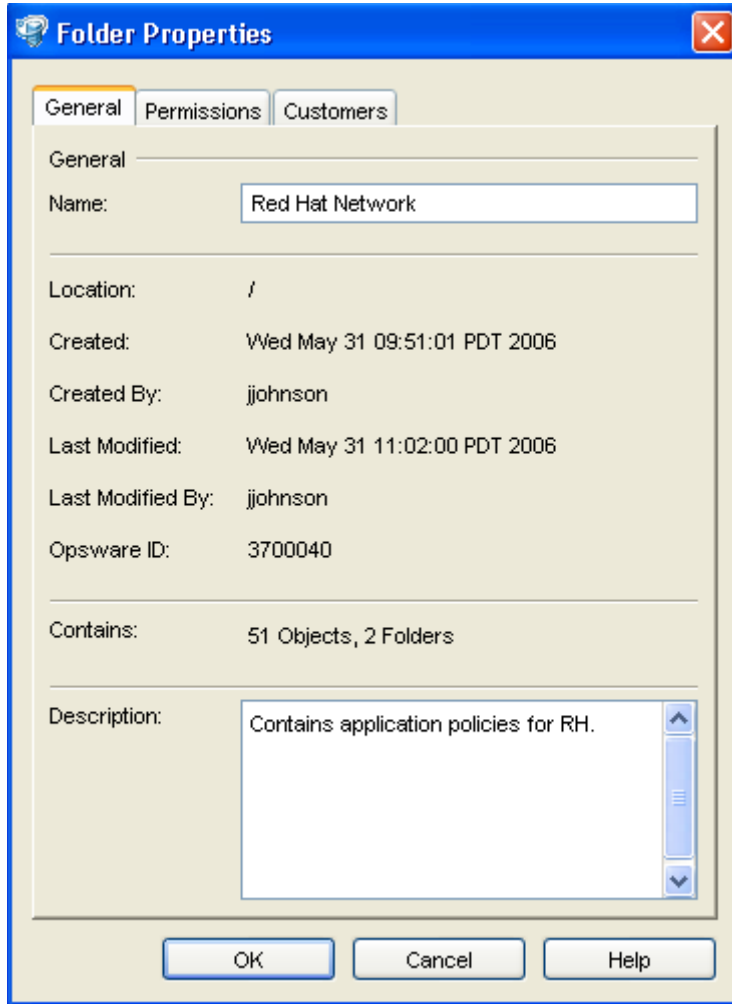
You can also set the Folder permissions and Customer permissions on the folder. See the *SA Administration Guide* for information about setting Folder and Customer permissions.

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

Perform the following steps to manage the properties of a folder in the SA 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.
- 3 From the **Actions** menu, select **Folder Properties**. The Folder Properties window appears as shown in [Figure 31](#).

Figure 31 Setting Folder Properties in the SA Client



- 4 Select the General tab to view the folder properties, such as the location of the folder in the Library, the SA ID associated with the folder, and the features and subfolders contained in the folder.
- 5 In the Name field, modify the name of the folder. In the Description field, enter a description of the folder.
- 6 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 SA administrator. See the *SA Administration Guide* for more information.

Perform the following steps to delete a folder in the SA 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 SA by uploading the packages to the Software Repository with the SA Client or by using the SA Command Line Interface (OCLI) Version 1.0.

The Software Repository provides a data store for all software that the SA 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 SA supports has a list of package types that you can upload. SA supports these package types on the supported operating systems, as shown in [Table 2](#).

Table 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, .I, .U, .lpp	N/A
	RPM	.rpm	N/A
	ZIP	.zip	N/A
	Application Installation Media	.zip	N/A
	Executable	.exe, .sh	N/A
HP-UX	Depot (contains products and filesets)	.tar, .depot	N/A
	ZIP	.zip	N/A
	Application Installation Media	.zip	N/A
	Executable	.exe, .sh	N/A
Linux	RPM	.rpm	N/A
	ZIP	.zip	N/A
	Application Installation Media	.zip	N/A
	Executable	.exe, .sh	N/A

Table 2 Supported Operating Systems and Package Types (cont'd)

Operating System	Package Type	File Formats	Additional Metadata
Solaris	Patch	.jar, .tar, tar.gz, .tar.Z, t.gz, .zip	N/A
	Patch Cluster (contains patches)	.tar, .tar.gz, tar.Z, .tgz, .zip	N/A
	Solaris package (contains package instances)	.pkg, .tar	N/A
	RPM	.rpm	N/A
	ZIP	.zip	N/A
	Application Installation Media	.zip	N/A
	Executable	.exe, .sh	N/A
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
	Application Installation Media	.zip	N/A
	Executable	.exe, .sh	N/A
OS Independent	Unknown	All	N/A

* For certain package types, SA 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, SA 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. SA automatically creates package entries for all the filesets that the LPPs contain. When viewing an LPP in the SA Client, users see which filesets it contains.

The 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 SA Client shows filesets and APARs in the Installed Packages list for a server.

LPP Metadata

SA 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



SA 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, SA 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 SA.

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.
- SA supports products, filesets, and patch products as installable software.



SA 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. SA does not support using HP-UX code words.

When a user uploads a depot, SA 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 SA 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. SA attempts to install it anyway. An error results if a newer version is already installed.



SA 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, SA does overwrite newer versions of filesets and products with older versions. SA does not support relocating packages for HP-UX.

Depot Metadata

SA 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.



For more information on creating a script to automate this step, see [Example: File – Script to Split a Depot by Product](#) on page 63 and [Example: File – Script to Split a Depot by Bundle](#) on page 63.

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 SA.

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 HP  
# 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 HP 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

SA 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, SA 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 SA, the software policies related to that RPM may be updated. See [Setting Installation and Update Options for an RPM Package](#) on page 47 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 SA.

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. SA adds package instances to and removes package instances from servers by using the `remediate` function. See the *SA User's Guide: Server Automation* for more information about `remediate`.

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

SA supports Solaris packages in the following ways:

- Users upload Solaris packages in the uncompressed data stream file format.
- SA can install interactive and non-interactive Solaris package instances. Interactive Solaris package instances require response files.
- SA 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 Agent on managed servers includes its own admin file that it uses when installing Solaris package instances. The admin file that the Agent uses is only used by SA 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 with a non-zero exit code 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.

SA supports response files in the following ways:

- Users create response files outside of SA by using the `pkgask` Solaris utility.
- By using the Solaris Instance Package Properties page in the SA 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 SA 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, SA 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 SA Client, SA might report an error when any server is remediated with that Solaris package instance.

When a user uploads a Solaris package, SA 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

SA 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 SA 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

SA 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

SA supports .msi files as installable software.

MSI Package Metadata

SA 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

SA 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 Agent Installer command line.

See the *SA User's Guide: Server Automation* for more information about the steps to install an 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, SA requires the user to provide the version of the service pack. When you upload Hotfixes and Security Patches, SA requires the user to provide the operating system version and the patch type.

ZIP Packages

ZIP Package Support

The SA 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. SA 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.

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

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

Info-Zip Compatible ZIP Packages

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

Info-Zip Compatible Package Metadata

SA 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 SA 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 SA, 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 SA Client or by using the SA 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).

SA converts the script contents from the UTF-8 encoding to the encoding that you select. Internally, SA 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 SA 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 Packages

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

As of SA 7.80, you can import multiple packages simultaneously. If a package that is being uploaded already exists in the Software Repository, SA allows you to replace (overwrite) the contents of the existing package, skip the package import (useful when importing multiple packages), or cancel the import. If you upload Solaris patch clusters that contain patches that already exist in the Software Repository, the patches are overwritten. However, SA preserves any reboot options or flags set for the patches in the SAS Web Client.



You must have a specific set of permissions to import packages. To obtain these permissions, contact your SA administrator. See the *SA 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.

- 2 From the **Actions** menu, select **Import Package**. The Import Software window appears.
- 3 Click **Browse** to locate and select the packages to import.
- 4 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 SA can extract the metadata contained in the package and correctly display the information in non-ASCII characters in the SA Client (for example, in the Package Properties pages). Package metadata includes comments, READMEs, scripts, descriptions, and content lists.

- 5 In the Import Software window, select the file type from the Filetype drop-down list. Some of the package types include Windows MSI, ZIP, Executable, application installation media, RPM, Solaris Package. When importing multiple files, all file must be of the same type.
- 6 Click **Browse** to specify the folder location for the package. The Select Folder window appears. You cannot specify the folder location for Solaris Patches and Solaris Patch Clusters since they are not located in folders.
- 7 From the Platform drop-down list, select the operating system family or operating systems. You can also select multiple operating system families.
- 8 Click **Import**.
- 9 If one of the packages you are importing already exists in the folder, you will be presented with a dialog that displays the file(s) with the same name as the one you are importing. You have the following options:
 - **Replace**: Replace (overwrite) the contents of the existing file.
 - **Replace All**: When there are multiple existing files with the same name as the file you are importing, you can replace (overwrite) the contents of all the existing file.
 - **Skip**: Skip the replacement of a single file. If you have multiple existing files with the same name as the file you are importing, you can select which files to skip or not. Skipping the import of a file does not affect other files with different names if you are importing multiple files. Only the specified file(s) will be skipped, the other specified files will be imported.
 - **Skip All**: All specified files with the same name as the file you are importing will be skipped and not replaced.
 - **Cancel**: Cancels the Import Package operation entirely. No file are imported.
 - **Help**: Provides online help for the current dialog.

Importing Application Installation Media

SA allows you to import a software application such as Symantec Antivirus, provided by a software vendor using the SA Client and then deploy the software application by using software policies. See the *SA User's Guide: Application Automation* for information about installing software.

Before importing a software application from the software vendor using the SA Client you are required to perform the following steps:

- 1 Obtain the application installation media from the software vendor on a CD or DVD, or download the application installation media.
- 2 Develop any required scripts or response files for the application media. The application must support silent install.

- 3 Create a ZIP file containing the application installer.

Once the ZIP file containing the application installer is created, you can use the SA Client to import the application installer to SA.



You must have a set of permissions to import application installation media. To obtain these permissions, contact your SA administrator. See the *SA Administration Guide* for more information.

Perform the following steps to import a application installation media:

- 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.
- 2 From the **Actions** menu, select **Import Package**. The Import Software window appears.
- 3 Click **Browse** to locate and select the packages to import.
- 4 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 SA can extract the metadata contained in the package and correctly display the information in non-ASCII characters in the SA Client (for example, in the Package Properties pages). Package metadata includes comments, READMEs, scripts, descriptions, and content lists.
- 5 In the Import Software window, select Application Installation Media from the Filetype drop-down list.
- 6 Click **Browse** to specify the folder location for the packages. The Select Folder window appears.
- 7 From the Platform drop-down list, select the operating system family or operating systems. You can also select multiple operating system families.
- 8 Click **Import**.
- 9 From the Navigation pane, select **Library > By Type > Packages**. The packages organized by operating system appear in the Content pane. Select the package and open.
- 10 In the Packages window, select Properties from the View pane. Enter the installation script and uninstallation script.
- 11 Select **Save** from the File menu.

Importing Executables

Perform the following steps to import executables:

- 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.
- 2 From the **Actions** menu, select **Import Package**. The Import Software window appears.

- 3 Click **Browse** to locate and select the packages to import.
- 4 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 SA can extract the metadata contained in the package and correctly display the information in non-ASCII characters in the SA Client (for example, in the Package Properties pages). Package metadata includes comments, READMEs, scripts, descriptions, and content lists.
- 5 In the Import Software window, select Executable from the Filetype drop-down list.
- 6 Click **Browse** to specify the folder location for the packages. The Select Folder window appears.
- 7 From the Platform drop-down list, select the operating system family or operating systems. You can also select multiple operating system families.
- 8 Click **Import**.
- 9 From the Navigation pane, select **Library ► By Type ► Packages**. The packages organized by operating system appear in the Content pane. Select the package and open.
- 10 In the Packages window, select Properties from the View pane. Enter the install command and uninstall command.
- 11 Select **Save** from the File menu.

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 SA 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:

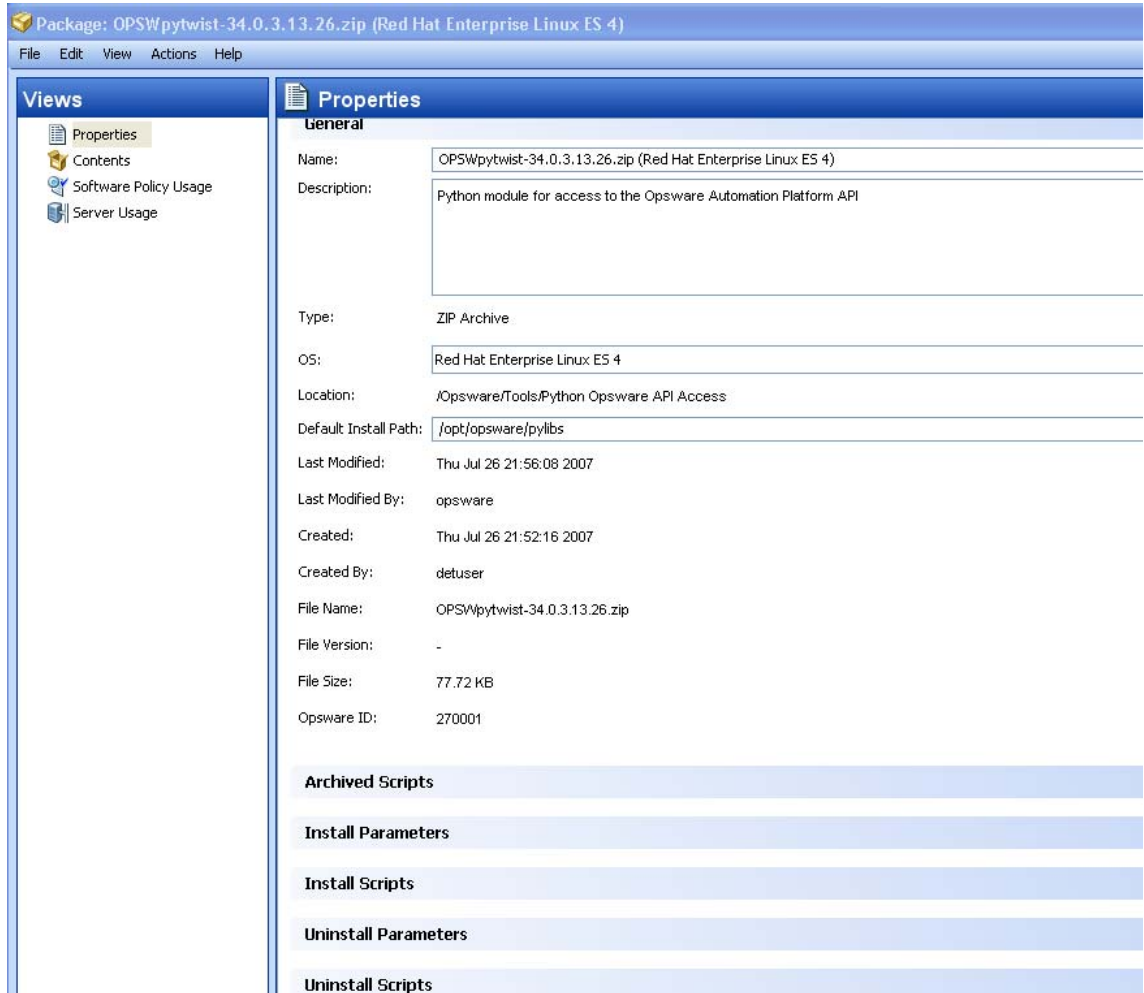
- 1 From the Navigation pane, select **Library** ► **By Type** ► **Packages**. The packages organized by operating systems appear in the Content pane.

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 as shown below.

Figure 32 Package Window



- 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 SA user who last modified the package.
- **Created:** The SA 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.
- **SA ID:** The unique SA 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:** (Only for Executables) The command that will be used to install the package. For executable packages you are required to enter the install command. The install command includes:

(Windows): Replace with install command, for example, start /wait “Description” “%EXE_FULL_NAME%” <arguments>

(UNIX): Replace with install command, for example, “%EXE_FULL_NAME%” <arguments>

The environment variable EXE_FULL_NAME will contain the fully qualified path to the executable when the Install command is run.

- **Install Flags:** (Only for RPM, MSI, Build Customization Scripts) The optional arguments to be run when the package is installed on a managed server.
- **Temporary Path:** (Only for application media) The temporary directory where the zip package is downloaded.
- **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

- **Install Script:** (Only for Application Installation Media) A script required to perform a silent installation of the application.
- **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. For executable packages you are required to enter the uninstall command.

- **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

- **Uninstall Script:** (Only for Application Installation Media) A script required to perform a silent uninstallation of the application.
- **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 SA Client.

You can edit a package's name, description, operating system association of the package, install parameters, install scripts, uninstall parameters, and uninstall scripts.



You must have a set of permissions to edit the package properties. To obtain these permissions, contact your SA administrator. See the *SA 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.
 - **OS:** The operating systems associated with the package.
 - **Location:** The location of the package in the folder hierarchy.
 - **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 back slashes.

- **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 Command:** (Only for Executables) The command that will be used to install the package. For executable packages you are required to enter the install command. The install command includes:

(Windows): Replace with install command, for example, start /wait "Description" "%EXE_FULL_NAME%" <arguments>

The Windows Start command can behave differently if the filename to run contains spaces. It is recommended that you always include a command Description as shown in the above example.

(UNIX): Replace with install command, for example, EXE_FULL_NAME% " <arguments>
- **Temporary Path:** (Only for application media) The temporary directory where the zip packages is executed.
- **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.
- **Install Script:** (Only for Application Installation Media): Specifies a script required to perform a silent installation of the application.
- **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 Command:** The command that will be used to uninstall the package.
- For executable packages you are required to enter the uninstall command.
- **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.
- **Uninstall Script:** (Only for Application Installation Media): Specifies a script required to perform a silent uninstallation of the application.
- **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 SA administrator. See the *SA 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

SA 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, SA determines the dependencies and installation order for the RPM package, and identifies if any conflicts exists between the dependencies. After you resolve the conflicts, SA installs the RPM packages on the managed server.

In the SA 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. SA also allows you to automatically download the Red Hat Linux Errata into SA and convert them to software policies. See [Automatically Importing Red Hat Network Errata](#) on page 85 for more information.

RPM Deployment Process

Deploying RPM packages on Linux servers involves the following steps:

- Uploading RPM packages. See [Importing Packages](#) on page 69 uploading packages for more information.
- Creating a software policy. See [Creating Software Policies and Software Templates](#) on page 39 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 [Setting Installation and Update Options for an RPM Package](#) on page 47 for more information.
- Attaching the software policy to a managed server. See the *SA User's Guide: Application Automation* for information about attaching a software policy to a server.
- Remediating the server against the software policy. See the *SA 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, SA 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. SA 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, SA installs the RPM packages specified in the software policy. See the *SA 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, SA will only install the latest version of the RPM package and its dependent packages.



During remediation SA 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](#)

SA allows you to install and update RPM packages on a server using software policies. After you import RPM packages to SA, you can add the RPM packages to the software policy. See [Importing Packages](#) on page 69 and [Setting Installation and Update Options for an RPM Package](#) on page 47 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 Criteria
- Auto Update Policy

The Install Criteria option determines whether the RPM package listed in the software policy will be installed on the managed server. The Auto Update Policy option 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 81 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 82 for more information.

- **Install Criteria:** If the **Install RPM always** option is selected, SA will install the RPM packages specified in the software policy on the managed server when you remediate the software policy on the managed server.

If you select the **Install RPM if only an earlier version is installed** option is selected, SA 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, SA does not install the RPM version specified in the software policy.

See [Setting Installation and Update Options for an RPM Package](#) on page 47 for more information on how to set these options.

Automatically Updating RPM Packages in a Software Policy

SA 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 or Release:** This option allows you to automatically update the version or release of the RPM packages in a software policy. If you select this option, SA 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.

▶ SA 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 this option, SA 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.

▶ SA 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 47 for more information 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 SA, you can set the set the Upgrade option for the RPM package in the Package properties Window. See [Editing Package Properties](#) on page 76 in this chapter for more information.

In the Package Properties window if you set the Install Mode option to Upgrade, then during remediation, SA 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 Install Mode option to Install, then during remediation SA 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 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
In the Install Criteria, the Install RPM only if an earlier version is installed option is selected	The Install Mode option is set to Upgrade	If a previous version of the RPM package is installed on the server, then SA 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 SA will not install the RPM version specified in the software policy.
In the Install Criteria, the Install RPM always option is selected	The Install Mode option is set to Upgrade	If the RPM package is not already installed on the managed server, then SA 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 SA 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.

Table 3 Setting Options for an RPM Package (cont'd)

Option selected for an RPM Package in the Software Policy window	Option selected for an RPM Package in the Package Properties Window	Action Taken
In the Install Criteria, the Install RPM only if an earlier version is installed option is selected	The Install Mode option is set to Install	If a previous version of the RPM package is installed on the server, then SA 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 SA will not install the RPM version specified in the software policy.
In the Install Criteria, the Install RPM always option is selected	The Install Mode option is set to Install	If the RPM package is not already installed on the managed server, then SA 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 SA will not install the newer version of the RPM package as specified in the software policy on the server.

In SA, 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 SA 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

SA 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 *SA User's Guide: Application Automation* for information on how to detach a software policy from a server.

When you remediate the server, SA 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, SA also uninstalls any RPM packages which depend on the RPM packages being uninstalled.

SA 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 *SA User's Guide: Application Automation* for more information about detaching a software policy.
- 2 Remediate the server to uninstall the RPM package. See the *SA User's Guide: Application Automation* for more information about remediating a server.
- 3 Create a new software policy. See [Creating Software Policies and Software Templates](#) on page 39 in this chapter for information about creating a software policy.
- 4 Add the older version of the RPM package to the software policy. See [Setting Installation and Update Options for an RPM Package](#) on page 47 in this chapter for more information adding RPM packages.
- 5 Attach the software policy to the server. See the *SA User's Guide: Application Automation* for more information about attaching a software policy.
- 6 Remediate the server to install the RPM package. See the *SA 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 *SA 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 SA. Using SA 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 SA Client.

Red Hat Network also consists of channels that contain packages. Using SA 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 SA Client

The `rhn_import` CLI program provided by SA 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, SA downloads the latest packages from the Red Hat Network

channel and then imports the packages to the Library in the SA Client and creates a channel based software policy. Thus, a channel based software policy reflects a particular channel. In the SA 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 SA Client](#) on page 86 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, SA downloads the latest packages from the Red Hat Network errata and then imports the packages to the Library in the SA 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 SA Client. In the SA 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 SA Client](#) on page 86 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, SA downloads the latest packages from the Red Hat Network erratum and then imports the packages to the Library in the SA 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 (part of the Slice Component bundle) and run the `rhn_import` program located in the following directory:

```
/opt/opsware/rhn_import/bin/rhn_import
```



Importing RPM packages from the Red Hat Network to SA 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. It is recommended 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/opsware/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 HP. The Configuration file provided by HP with the `rhn_import` program is located in the following directory:

```
/etc/opt/opsware/rhn_import/rhn_import.conf
```

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

The `rhn_import` program, allows you to create errata-based, erratum-based and channel-based software policies in the SA Client. After successfully running the program, you can view the properties of errata-based, erratum-based, and channel-based software policies in the SA Client. You can view properties such as the SA 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 HP ID of the software policy. HP recommends that you do not edit the software policies which have been created by the `rhn_import` program.

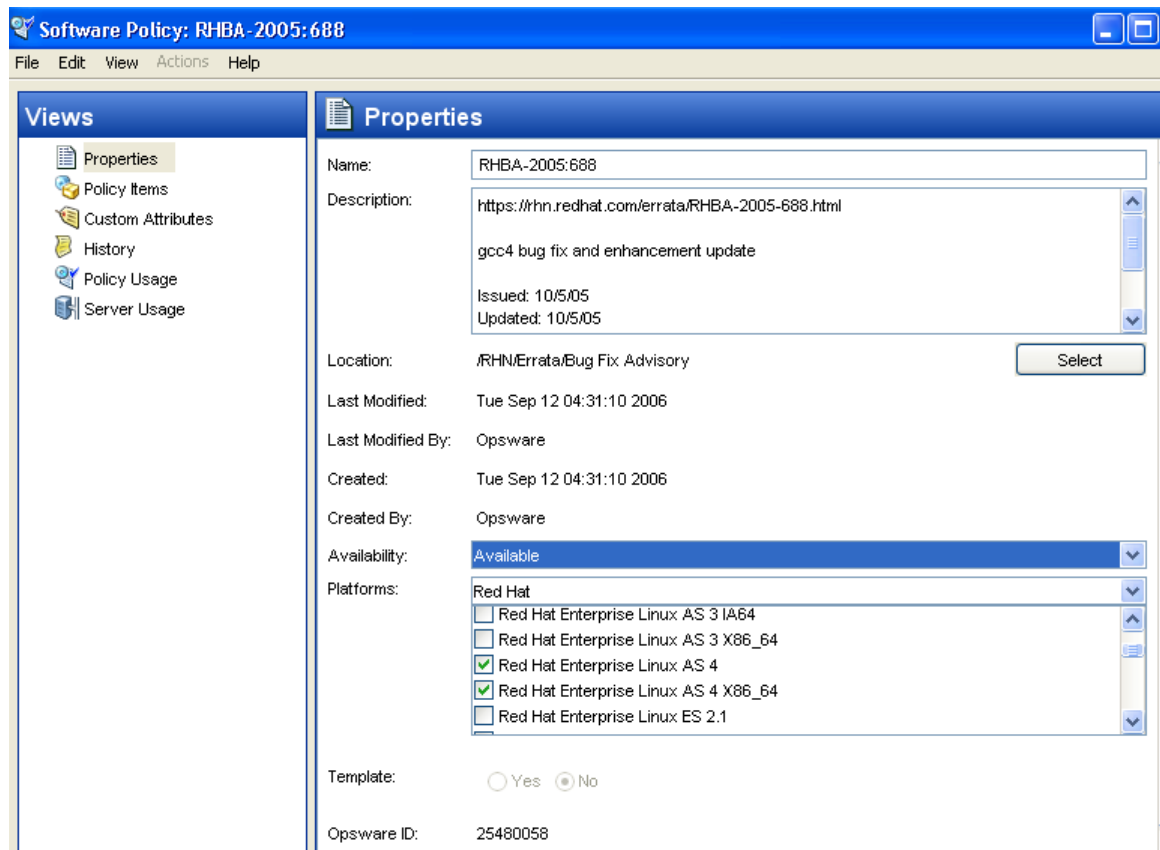


You must have a set of permissions to manage errata-based, erratum-based, and channel-based software policies. To obtain these permissions, contact your SA administrator. See the *SA Administration Guide* for more information.

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 33](#).

Figure 33 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 HP to create the software policy.
 - **Last Modified:** Corresponds to the time when the errata based software policy was modified.
 - **Availability:** Contains the HP 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.

3 Operating System Provisioning Setup

This chapter contains information about setting up SA OS Provisioning, a feature that allows you to use the SA 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 SA administrator for information about the installation and configuration of SA 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 also need to prepare an operating system installation profile for each OS. See [Overview](#) on page 103 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 SA administrator. For more information about which permissions are required, see the *SA 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 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 108 or [Defining an OS Installation Profile — Windows](#) on page 112 for more information.
- Create an OS sequence in the SA 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 94.
- 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 101.
- 3 Create a Solaris profile with a text editor. See [Sun Solaris Profiles](#) on page 105.
- 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 108.

- 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 126.
- 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 135.
 - [Custom Attributes for Sun Solaris](#) on page 136.
- 7 Create an OS Sequence in the SA 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. 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 94 and [Creating an MRL with the Import Media Tool](#) on page 101.
- 2 Create a configuration file with a text editor. See the following sections:
 - [Red Hat Linux Configuration Files](#) on page 105
 - [VMware ESX Configuration Files](#) on page 105
 - [SUSE Linux Configuration Files](#) on page 106
- 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 108.
- 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 129
 - [VMware ESX Build Customization Scripts](#) on page 130
 - [Requirements for Linux Build Customization Scripts](#) on page 129.
- 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 135
 - [Custom Attributes for Linux or VMware ESX](#) on page 137

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 150.

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 94.
- 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 101.
- 3 Create a Windows response file with a text editor. See [Microsoft Windows Response Files](#) on page 106.
- 4 Prepare an OS installation profile for the Windows OS in the SAS Web Client. See [Defining an OS Installation Profile — Windows](#) on page 112. 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 111.

- 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 135.
- [Custom Attributes for Microsoft Windows](#) on page 139.

- 8 (Optional) Create a Windows boot floppy. See [Creating a Windows Boot Image](#) on page 148. You 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. SA 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 146.

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 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/CIFS for Windows. After copying the OS media to the Media Server, you must “import” it into SA by running the Import Media tool (a utility script included with SA). 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 automatically-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 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 SA. An MRL should resolve to the Media Server in the local facility where SA 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, SMB, or CIFS.
- Detects the OS (Solaris, Linux, VMware ESX, or Windows) and version of the media.
- Creates that MRL in SA 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 SA Client. These packages can be added to a software policy in the SA 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.

As of SA 7.80, the `import_media` utility no longer modifies the media during new Linux/Windows media import.

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 SA administrator for the host name of the Media Server.)

You must know what locations were specified for the OS media. When SA was installed, the HP BSA Installer prompted for the path names of the root directories for the Windows, Solaris, Linux, and VMware ESX OS media on the Media Server. If necessary, contact your SA 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 SA user credentials (username and password). This enables you to use the import media tool to create an MRL and upload packages. Unless specified in the import media argument, you are 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.

Task 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 SA installation. If necessary, contact your SA administrator for the locations of the OS media root directories.

Task 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 SA installation. If necessary, contact your SA administrator for this information.

- **Microsoft Windows Server Server 2003/2008 x64:**

- Copy the OS media files to the location on the Media Server specified during the SA installation.

For Windows Server 2003 32-bit, the import media command would look like this:

```
import_media smb://<SMB Server>/<share>/<path>/i386
```

For Windows Server 2003/2008 64-bit, the import media command would look like this:

```
import_media smb://<SMB Server>/<share>/<path>
```

If necessary, contact your SA 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]
```

- **Red Hat Linux or VMware ESX:**

- a Download the Red Hat Enterprise Linux 5 images to the Core.
- b Connect to the Core as root using ssh (you will need to run `mount` commands).
- c Create a temporary folder for loop mounting the images.
- d Create a directory under the media server's Linux media path. The Linux media path is a NFS share configured during the core install.
- e Loop mount the first image:

```
mount -o loop rhel-5-server-i386-disc1.iso <tmp_mount_dir>
```

- f Change to the temporary directory

```
cd <tmp_mount_dir>
```

- g Issue the command

```
tar cf - . |(cd /media/hp/linux/RHEL5-Server/ && tar xfps -)
```

- h cd out of the temporary directory.

- i Unmount the temporary directory:

```
umount <tmp_mount_dir>
```

- j Repeat steps from 5 to 9 for the remaining 4 images.

- k You can now import the media.

- **Solaris - Importing Windows Media**

- When you launch the `import_media` utility from a Solaris server, you must use SMB to import Windows media.

- **Linux - Importing Windows Media:**

- When you launch the `import_media` utility from a server running Red Hat 5 kernel or higher, you must use CIFS to import Windows media. The syntax is the same as Samba, such as:

```
import_media cifs://<SMB Server>/<share>/<path>/i386
```

You can use either SMB or CIFS to import Windows media for all other Linux kernel versions.

- **SUSE Linux:**

Suse Linux 9:

- a Create the following directory structure:

```
sles9
sles9/suse
sles9/suse/CD1
sles9/core
sles9/core/CD1
sles9/core/CD2
sles9/core/CD3
sles9/core/CD4
sles9/core/CD5
yast
```

- b Copy the contents of the first Suse Linux 9 CD1 to the `sles9/suse/CD1` directory.



The directory numbering does not match the CD numbering which can be confusing, so be sure you are copying the contents of the CDs into the correct directories

- c Copy the contents of the second Suse Linux 9 CD2 to the `sles9/core/CD1` directory.

- d Copy the contents of the third Suse Linux 9 CD3 to the `sles9/core/CD2` directory. Continue this sequence until all the CDs have been copied to their respective directories.

- e In the `sles9` directory create the following symbolic links:

```
ln -s sles9/suse/CD1/boot boot
ln -s sles9/suse/CD1/media.1 media.1
ln -s sles9/suse/CD1/content content
ln -s sles9/suse/CD1/control.xml control.xml
```

- f Using an editor, create the `instorder` file in the `yast` directory. It should contain the following information:

```
/suse/CD1
/core/CD1
```

- g Using an editor, create the `order` file in the `yast` directory. It should contain the following information:

```
/suse/CD1      /suse/CD1
/core/CD1      /core/CD1
```

- **Suse Linux 9 with Support Pack:**

You will need all nine Suse CDs, three contain the Support Pack and six FCS CDs. Follow the standard installation steps above first then complete the following tasks:

- a Add the following directories:

```
sles9/sp3/CD1
sles9/sp3/CD2
sles9/sp3/CD3
```

- b Copy the contents from the SP3 CD1, CD2, and CD3 to `sles9/CD1`, `sles9/CD2`, and `sles9/CD3`, respectively.

- c Modify the `instorder` and `order` files to include the `sp3` directory you added in the preceding step *at the top of each file*.


```
instorder
/sp3/CD1
/suse/CD1
/core/CD1
```

```
order
/sp3/CD1    /sp3/CD1
/suse/CD1   /suse/CD1
/core/CD1   /core/CD1
```

- d Log on as root to the repository server and create the following additional symbolic links:

```
ln -s sp3/CD1/driverupdate driverupdate
ln -s sp3/CD1/linux linux
```

- **Suse Linux 10:**

As of Suse Linux 10 it is no longer necessary to use the above procedures. You can install everything into a single directory.

More Suse Linux Information:

For more information on SUSE linux installations, see:

<http://www.suse.com/~ug/>

http://www.suse.com/~ug/autoyast_doc/index.html

For more information on AutoYaST Module development, see:

http://www.suse.com/~ug/autoyast_doc/devel/index.html

For more information about development and documentation links for AutoYaST for SLES 9 & 10, see:

<http://developer.novell.com/wiki/index.php/YaST>

For AutoYaST Documentation from OpenSUSE, see:

http://en.opensuse.org/YaST_Autoinstallation

If required, for information on how to deal with multiple sources, see

http://www.suse.com/~ug/autoyast_doc/index.html

- **Sun Solaris**

- a Download the Solaris 10 images to the Core.
- b Connect to the Core as root using ssh (you will need to run `mount` commands).
- c Create a temporary folder for loop mounting the images.
- d Create a directory under the media server's Linux media path. The Linux media path is a NFS share configured during the core install.
- e Loop mount the first image:

```
mount -o loop sol-10-u4-ga-x86-v1.iso <tmp_mount_dir>
```
- f Change to the temporary directory:

```
cd <tmp_mount_dir>
```
- g Issue the command:

```
tar cf - . |(cd /media/hp/sunos/Solaris10/ && tar xfps -)
```

- h cd out of the temporary directory.
- i Unmount the temporary directory:
umount <tmp_mount_dir>
- j Repeat steps from 5 to 9 for the remaining 4 images.
- k You can now import the media.

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 Media Server.

To create these directories on the Media Server, perform the following tasks:

- 1 Log on to Media Server as root.
- 2 Navigate to `Windows_media_share/i386` and create the following directory:
`OEM/$1/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:

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

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

Windows Media: Hosting Windows Media on a Windows 2000 Server using a Share

You want to host your Windows media on a Windows 2000 server using a share and have access to the share is available to a local user on the server. For example:

```
Server / Share:  
\\servername\IOP
```

`user: username password: userpassword` is used to mount the share. SA Windows build script directories have the user hardcoded to `guest` with no password. Many security policies do not allow for an enabled `guest` account, read-only share.

Perform the following tasks to set up the share:

Edit the file:

```
/opt/hp/buildscripts/windows/buildserver.py
```

and replace these lines:

```
system_ini["network"]["username"] = self.mrl_username  
system_ini["network"]["logondomain"] = self.mrl_domain  
system_ini["network"]["workgroup"] = self.mrl_domain
```

with your share credentials. Also edit the following lines specifying the correct username/ password:

```
# formulate net logon command line
logonCmd = []
logonCmd.append("lh %ramdrv%\mslanman\%net")
logonCmd.append("logon")
logonCmd.append(self.mrl_username)
logonCmd.append(self.mrl_password)
```

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 (Slice Component bundle host) 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
- Window CIFS:
cifs://<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.

When using 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: \$ % & ' _ @ { } ~ ` ! # (). 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 SA was installed to write-protect the Windows media share. The Import Media tool prompts for the password each time you run it. Contact your SA administrator for this password

Table 4 lists the command line options available for the import media command.

Table 4 Import Media Tool Command Line Options

Import Media Tool Option	Description
--help	Display this help.

Table 4 Import Media Tool Command Line Options (cont'd)

Import Media Tool Option	Description
<code>--folder</code>	Override Folder location. Default is: “/Package Repository/OS Media/<Platform Name>”.
<code>--medianame=<displayname></code>	Override automatically-generated display name. Note: Use '_' to escape spaces in the name.
<code>-hpsa-username</code>	Username for authenticating to SA. If you do not supply <code>-hpsa-username</code> on the command line, you are prompted to enter it. If you do not have a valid SA user name and password, contact your SA administrator.
<code>-hpsa-password</code>	Password for the SA username. Warning: This option is not recommended, since passing passwords as command line options is insecure. When this option is omitted, the user is 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”.
<code>--smbpasswd=<password></code>	Use this password for SMB access. Note: This appears 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 is prompted for the password securely.
<code>--logfile=<logfile></code>	Override log file location. Default is: <code>/var/log/hp/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 SA 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 (Slice Component bundle) 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 where you uploaded the OS media on the OS Media Server.

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) SA name for the platform detected in the media being imported. If the folder does not exist, then it is 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.

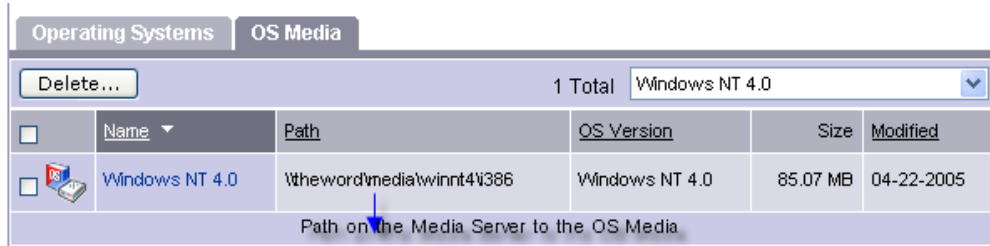
For information on the command line options for the Import Media tool, see [Import Media Tool Syntax and Options](#) on page 99.

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 34](#).

Figure 34 OS Media Page in the SAS Web Client



- 4 Click the display name for the MRL that you want to edit. The **Edit OS Media** page appears, as [Figure 35](#) shows.

Figure 35 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:	ifs://mediaserver.c76.dev.opsware.com/media/ops
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

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 108 or [Defining an OS Installation Profile — Windows](#) on page 112 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 90 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 93 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 104 for more information.)
 - A build customization script, which can modify the installation process at certain points. (See [Build Customization Scripts](#) on page 122 for more information.)
 - *Microsoft Windows Only*: a Hardware Signature, which contains hardware specific information. ([Hardware Signature Files for Windows](#) on page 111 for more information.)

[Table 5](#) compares the installation resources across operating systems.

Table 5 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 5 Installation Resources for OS Installation Profiles (cont'd)

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 116 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 SA 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 enables a firewall, you must ensure that all necessary ports and protocols for communication between the SA core and the OS Build Agent and the SA Agent are allowed. Refer to the *SA Planning and Installation Guide* for details. To help isolate firewall related issues, you should leave firewalls disabled while configuring OS provisioning

for the first time and reenabling them once the system is correctly configured. For Red Hat EL 4 and Red Hat EL 5, the following line in your `ks.cfg` profile will enable the firewall and allow the SA Agent to function correctly:

```
firewall --enabled --port 1002:tcp,1002:udp,1001:tcp,1023:tcp
```

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, SA 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, and so on.

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, SA 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 may 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, SA 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 107 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=HP
  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=XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Using OS Installation Profiles

This section discusses the following topics:

- [Defining an OS Installation Profile — Unix](#)
- [Hardware Signature Files for Windows](#)[Defining an OS Installation Profile — 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 SAS Web Client or from the SA 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**.
 - *SA 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 36](#) shows.

Figure 36 Describe OS Page in the Prepare Operating System Wizard

The screenshot shows a web interface for the 'Prepare Operating System' wizard. The main heading is 'Describe OS'. Below the heading, there is a sub-heading 'Describe OS' and a prompt: 'Enter the following information to describe the operating system.' The form contains the following fields:

- Name:** A text input field.
- Description:** A text area with scrollbars.
- Customer:** A dropdown menu with 'Customer Independent' selected.
- OS Version:** A dropdown menu with 'OS Independent' selected.

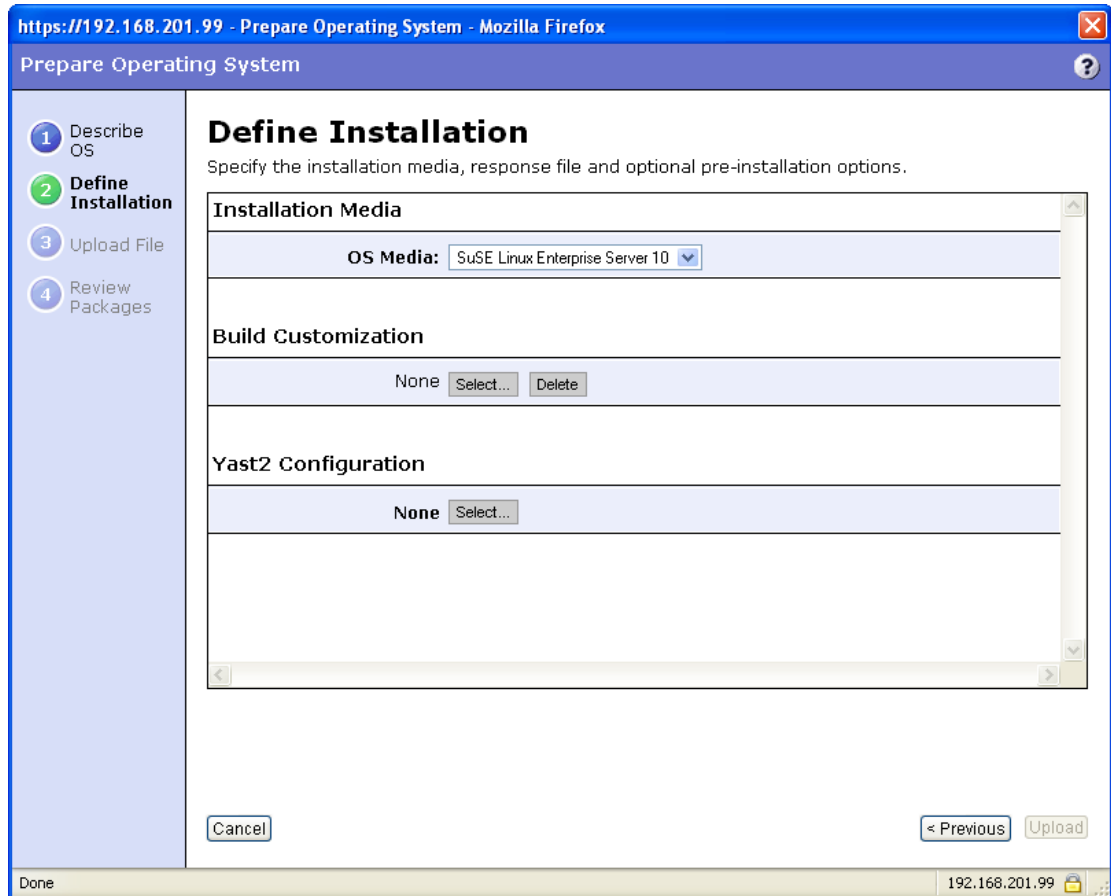
At the bottom of the form, there are three buttons: 'Cancel', '< Previous', and 'Next >'. On the left side, there is a vertical navigation pane with four steps: 1 Describe OS (highlighted), 2 Define Installation, 3 Upload File, and 4 Review Packages.

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 SA 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 37](#) shows.

Figure 37 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 93 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 appears in the popup window after you upload them through the SAS Web Client, see [Using Build Customization Scripts](#) on page 122.)

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**.

SA 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 setup response File (`unattend.txt`) typically contains a mix of generic operating system configuration settings and hardware-specific driver configuration settings. This mixture of generic and hardware-specific configuration settings can make it difficult to manage a single OS Installation Profile that must be used by many different hardware models.

SA includes a mechanism called *Hardware Profiles* that allow you to keep the generic configuration settings in `unattend.txt` separate from the hardware-specific driver configuration settings.

During OS provisioning, SA will examine the server being provisioned and, if a matching Hardware Profile is available for the server model, will automatically add in the appropriate hardware-specific driver configuration settings from `unattend.txt`.

Based on the hardware you expect to provision, you can upload hardware-specific files for each Windows OS Installation Profile. You can then 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 to be provisioned.

Utilities referenced by the Hardware Signature file must be accessible through the network during build time.

Example Hardware Signature File

The following is an example of a Hardware Signature file that would be used for installing Windows XP on a VMWare ESX guest with an LSI Logic SCSI controller:

```
;Windows Setup Answer File
;Validated for use with HP
;Goal with this file is to leave things unspecified as much as
;possible, therefore taking all the defaults
;Only including the absolutely essential directives for full ;unattended
operation

;Version 1.0 Jun 02 2007
;Contact Peter Lyons <plyons@hp.com> with questions

;-----
;KNOWN TO WORK WITH THE FOLLOWING SETUPS
;-----

;SA 7.0 (dev version), DOS 7.1 network boot image
;Windows XP Pro SP2 media
;VMware ESX 3.0.1 guest configured for Windows XP
;with a LSI Logic SCSI controller
;(Nota Bene BusLogic is the default in the ESX guest setup ;wizard)
;512 MB RAM, 1 NIC, 2 CPU

[GuiUnattended]
AdminPassword=hp
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]
DriverSigningPolicy=Ignore
ExtendOemPartition=1
FileSystem=ConvertNTFS
OemPnPDriversPath=Drivers\NIC
OemPreinstall=Yes
OemSkipEula=Yes
TargetPath=*
UnattendMode=FullUnattended

[UserData]
ComputerName=*
;FullName=HP
;OrgName=HP
;This is the HP volume license
;You can/should also set this as a custom attribute
;"ProductKey"
;on the OS Installation Profile
ProductKey=MWDGH-CKXY9-6H9T2-GHVFK-DVWHG

```



The use of Hardware Signatures files is not required for Sun Solaris or Red Hat Linux operating systems because Solaris and Linux distributions do not need to be specifically 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**.
 - SA 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 38](#).

Figure 38 Describe OS Page in the Prepare Operating System

Prepare Operating System

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 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 SA supports).
- **Description:** *(Optional)* Provide a text description to identify the platform and hardware support.

- 3 Click **Next**. The Define Installation page appears, see [Figure 39](#).

Figure 39 Define Installation Page in the Prepare Operating System Wizard

The screenshot shows a web browser window titled "Prepare Operating System" with the URL "https://192.168.201.3 - Prepare Operating System - Mozilla Firefox". The main content area is titled "Define Installation" and includes the instruction "Specify the installation media, response file and optional pre-installation options." The form is divided into 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, there are "Cancel", "< Previous", and "Upload" buttons. 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".

- 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 93 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.



For Windows Server 2008 provisioning, you must use WinPE.

When a server is booted with the WinPE pre-installation environment, it appears in the Server Pool in the SAS Web Client and in the Unprovisioned Servers list in the SA 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 are shown in Figure 40.

Figure 40 Default Values used for WinPE Installation Options in OS Installation Profile

The screenshot shows a window titled "Installation Options". At the top, there are two radio buttons for "Select Installation Type": "DOS" (unselected) and "WINPE" (selected). Below this, there are two text input fields. The first is labeled "Custom Disk Partitioning:" and contains the following text: `rescan`, `select disk 0`, `clean`, `create partition primary`, `active`, and `assign letter=C`. The second is labeled "Custom Disk Formatting:" and contains the text: `format.com C: /FS:NTFS /Q /Y /V:`. At the bottom, there is a label "Install Drive:" followed by a text input field containing the letter "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 122 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**.

SA 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 SA Client. (See [Default Custom Attribute Values](#) on page 135. For information on how to set custom attributes for software policies, see [Adding Custom Attributes to OS Installation Profile \(SAS Web Client\)](#) on page 141.
- Specifying custom disk partitioning and custom drive formatting (For Windows servers booted with WinPE).
- Setting up configuration tracking for an OS Installation Profile.

See the *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 41](#)). 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 SA Client.

Note that, you cannot change the customer association for an OS installation profile.

Figure 41 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 93 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 122 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 shown in [Figure 42](#).

Figure 42 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 are:

```
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 SA 7.80, you should add packages to an OS installation profile using software policies attached to OS sequences. This is because 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 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 SA 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 do 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 *SA User’s Guide: Application Automation* for more information.

The method described in this section is provided for those using versions of SA 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 43](#) shows.

Figure 43 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 <input type="button" value="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 SA 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 shown in [Figure 44](#).

Figure 44 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 SA Client, perform the following steps:

- 1 Launch the SA Client using one of the following methods:
 - From the **Power Tools** section of the SAS Web Client home page
 - From **Start** ► **All Programs** ► **SA 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 Red Hat/SLES 10 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 Process \(WinPE Boot Image\)](#)
- [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 SA 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 SA Client.

To import a build customization script into the SA 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 appear in a list when you click **Select**.

See [Defining an OS Installation Profile — Unix](#) on page 108 or [Defining an OS Installation Profile — Windows](#) on page 112 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 6](#) 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 6 Sun Solaris Build Process

Phase	Build Process Steps
Pre-installation	<ol style="list-style-type: none">1 A user boots the installation client over the network by entering the following command in a console attached to the server: <pre>boot net:dhcp - install</pre>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 begin script. The begin script is used to start the OS Build Agent.3 The OS Build Agent registers with the OS Build Manager.4 The Solaris build script probes the hardware configuration of the installation client and registers it with SA. The installation client then appears in the Server Pool list in the SAS Web Client.

Table 6 Sun Solaris Build Process (cont'd)

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 \$SI_PROFILE, the standard JumpStart location for dynamic JumpStart profiles.</p> <p>8 The Solaris build script executes the build customization script: /sbin/sh run Pre-JumpStart</p> <p>9 The Solaris build script validates the profile by using the JumpStart installer (pfinstall) in test mode.</p> <p>10 The Solaris build script causes the OS Build Agent to run in the background, allowing the JumpStart begin script to complete.</p> <p>11 The JumpStart installer pfinstall 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 OCC Client.</p> <p>12 The JumpStart installer pfinstall completes and runs the JumpStart finish 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: /sbin/sh run Post-JumpStart</p> <p>14 The installation client reboots.</p>
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: /sbin/sh run Pre-Agent</p> <p>17 The Solaris build script installs the SA Agent.</p> <p>18 The Solaris build script executes the build customization script: /sbin/sh run Post-Agent</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 *SA Content Migration Guide* for more information on how remediation works to install software on servers.

Solaris Provisioning from a Boot Server on a Red Hat/SLES 10 Linux Server

If you need to provision a Solaris server and the Boot Server is on a Red Hat/SLES 10 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 or NFS v4

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
```

To disable NFS v4 on a Red Hat Linux Boot Server host, 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 lines:

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

- 3 Restart NFS:

```
/etc/init.d/nfs stop  
/etc/init.d/nfs start
```

To disable NFS v4 on an SLES 10 Boot Server host:

- 1 On the Boot Server host, create the following file:

```
/etc/sysconfig/nfs
```

- 2 In the newly created `nfs` file, add the following line:

```
NFS4_SUPPORT="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 7](#) 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 7 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 standard Red Hat boot image and mounts the second stage image specified by the kernel parameters.3 Anaconda is replaced by a custom SA script that is used to invoke the OS Build Agent.4 The OS Build Agent registers with the Build Manager.5 The Linux build script probes the hardware configuration of the installation client and registers it with SA, causing the installation client to appear in the Server Pool list in the SAS Web Client.
Phase One	<ol style="list-style-type: none">6 In the SAS Web Client, a user selects the target version of Linux to install on the installation client.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.8 The Linux build script copies GRUB onto the partition and installs it into the MBR.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.10 If the Custom Attribute <code>kernel_arguments</code> is set for the OS installation profile, these kernel arguments are appended.11 The OS Build Agent exits and the server reboots.

Table 7 Linux Build Process (cont'd)

Phase	Build Process Steps
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 SA 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 boots from the newly installed OS.</p>
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 SA 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 ends.
- 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 SA 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 127.

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 8](#) 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 8 Microsoft Windows Build Process (DOS Boot Image)

Phase	Build Process Steps
Pre-installation	<ol style="list-style-type: none"> 1 A user boots an installation client over the network by using a PXE network bootstrap program or by using the Windows Boot Image. 2 The user selects a Windows boot type from the menu on the console for the PXE network bootstrap program. 3 PXE boots the Windows OS Build Agent over the network. 4 The OS Build Agent prompts the user to create a FAT boot partition on which to install Windows. 5 The OS Build Agent collects pertinent hardware information and registers the information with SA. <p>The server is ready to be provisioned and is available for selection from the Server Pool in the SAS Web Client.</p>
Phase One	<ol style="list-style-type: none"> 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. 7 If you specified a build customization script in the OS installation profile, the Windows build script runs the Pre-Copy hook. 8 The Windows build script mounts the Windows installation media as indicated by the Media Resource Location (MRL). 9 The Windows build script initiates a Windows unattended setup. 10 The Windows build script waits for a Windows unattended setup to complete and Windows to boot for the first time.
Phase Two	<ol style="list-style-type: none"> 11 Windows boots for the first time. 12 If you specified a build customization script in the OS installation profile, the Windows build script runs the Pre-Agent hook. 13 If a build customization script was specified in the OS installation profile, it is executed by the Windows build script. 14 The Windows build script installs the Agent. <p>The Windows build script exits and Phase Two is complete.</p>

Windows Build Process (WinPE Boot Image)



In order to perform PXE booting of a VMWare ESX Windows 2003 x86 or x86_64 VM using WinPE, the minimum required RAM is 512MB (higher than the VMWare recommended RAM minimum).

Table 8 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 9 Microsoft Windows Build Process (WinPE)

Phase	Build Process Steps
Pre-installation	<ol style="list-style-type: none"> 1 A user boots an installation client over the network by using a PXE network bootstrap program or by using the WinPE. 2 The user can install either WinPE x86 32 bit or WinPE x64 64 bit pre-installation environment. 3 PXE boots the Windows OS Build Agent over the network. When using the WinPE pre-installation environment, you will not be prompted to create a disk partition. 4 The OS Build Agent collects pertinent hardware information and registers the information with SA. 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"> 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. 6 The Windows build script mounts the Windows installation media as indicated by the Media Resource Location (MRL). 7 The Windows build script initiates a Windows unattended setup. 8 The Windows build script waits for a Windows unattended setup to complete and Windows to boot for the first time.
Phase Two	<ol style="list-style-type: none"> 9 Windows boots for the first time. 10 If a build customization script was specified in the OS installation profile, it is executed by the Windows build script. 11 The Windows build script installs the Agent. The Windows build script exits and Phase Two is complete.

Legacy Build Customization Script run.bat

In previous releases of SA, 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 is run only 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 SA 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.
- 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). 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.
- 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 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. To manually test this hook from a command shell, execute it using:

```
cmd /c run.cmd
```

which mimics the build agent environment as closely as possible (and prevents an “exit” in the script from causing an exit from 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 \(SA Client\)](#)

In addition to the customization provided by using build customization scripts, each build script uses custom attributes.

The SAS Web Client and SA 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 “Adding Custom Attributes to OS Installation Profile (SAS Web Client)” on page 141.

For OS provisioning, SA 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 \(SA Client\)](#) on page 142 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 10](#).

Table 10 Sun Solaris Custom Attributes

Keyword	Description
archive_location	NFS path to a Flash Archive (flar) to use instead of OS media. Example Value: <code>nfs://mediaserver.company.com/flars/sunos5.10_basic.flar</code>
boot_options	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> . Example Value: Values will vary, see your Solaris documentation.
reboot_command	The command the OS Build Agent uses to issue a reboot during Solaris SPARC reprovisioning. The custom attribute value is not the entire command, rather it is the next boot command for the Open Boot PROM. The full command is <code>/usr/sbin/reboot -l -- 'net:dhcp - install</code> , only <code>net:dhcp - install</code> is replaced by the <code>reboot_command</code> value. Example Value: <code>net2:dhcp - install</code>

Table 10 Sun Solaris Custom Attributes (cont'd)

Keyword	Description
root_password	<p>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>.</p> <p>If a value is not set, the system will not have a root password.</p> <p>Example Value: Field 2 from the <code>/etc/shadow</code> file</p>
timezone	<p>Sets the time zone for the configuration of the installation client (sets TZ 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 US/Pacific. You can also specify any valid Olson time zone.</p> <p>Example Value: Any value in the <code>/usr/share/lib/zoneinfo</code> directory on a solaris server.</p>
system_locale	<p>Sets the language for the configuration of the installation client (sets LANG 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 locale 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> <p>Example Value: "C", "en_US.UTF-8", "ja_JP.UTF-8".</p> <p>See http://developers.sun.com/dev/gadc/faq/locale.html</p>
required_patches	No longer supported.
nfsv4_domain	<p>Sets the system's default NFS version 4 domain name. This value is substituted into <code>/etc/default/nfs</code> next to <code>"NFSMAPID_DOMAIN=</code>.</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> <p>Example Value: <code>company.com</code></p> <p>See: http://docs.sun.com/app/docs/doc/816-4555/6maoquib2?a=view#rfsrefer-133</p>

Custom Attributes for Linux or VMware ESX

You can use custom attributes to specify additional arguments to the kernel where 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 11 Linux or VMware ESX Custom Attributes

Keyword	Description
boot_disk	Values: A raw device name without "/dev/" such as "sda", "hdc", "cciss/c0d1"
boot_kernel	Values: "rhel30", "rhel40", "rhel50", "rhelia" "rhel3ia", "rhel4ia" Note: This custom attribute is only used for reprovisioning. The value of this custom attribute specifies the type of kernel the server boots to during reprovisioning.
kernel_arguments	Values: "noapci", "root=LABEL=/", "quiet", "splash"
ksdevice	Values: "eth0", "eth1", etc. Note: This custom attribute is used in the Media Boot Client (MBC) to create a server record. The Server Browser of this device has the following custom attribute: <pre>kernel_arguments =ksdevice=eth1 ksdevice eth1</pre> When powering on this device and PXE booting it, you do not need to specify the kickstart device.

Using the boot_disk Custom Attribute to Specify the Boot Drive

For certain servers you may need to specify the correct boot disk using the boot_disk custom attribute. Table 11 describes the usage for the boot_disk custom attribute.

SA uses the values specified with the boot_disk custom attribute to determine which disk to partition, format, and install the Assisted Installer image on.



The device you select must be configured as the first internal boot device in the BIOS. If the value of the boot_disk custom attribute is not found to exist on the hardware, SA logs a message and reverts to the original disk selection logic.

Sample ks.cfg File

The boot_disk custom attribute requires certain modifications to your Kickstart file in order to function properly. The following is a sample ks.cfg file for use with Red Hat Linux AS 4:

```
#Red Hat Kickstart Answer File
#Validated for use with Opaware
#This file supports a non-default boot_disk

#VERSION: 1.1 20080804
```

```

auth
bootloader --driveorder=@.boot_disk@
clearpart --drives=@.boot_disk@ --initlabel
part / --ondrive=@.boot_disk@ --asprimary --size=500 --grow
part swap --asprimary --size=250 --ondrive=@.boot_disk@
keyboard us
lang en_US.UTF-8
langsupport --default en_US.UTF-8 en_US.UTF-8
reboot #require by OPSW
rootpw password
text
timezone --utc UTC
#Required for opsware
firewall --disabled
%packages
@base



```

%pre
#OK, the purpose of this is to initialize all partition tables
#If anaconda finds a completely new raw disk or any disk with an #invalid
partition table, it goes interactive. This makes sure
#anaconda continues unattended
for D in `sfdisk -l 2>/dev/null | grep "unrecognized partition" | cut -d : -f
1 | tr -d " "|xargs`
do
 echo "Found an uninitialized partition table on ${D} according to sfdisk.
Adding a new empty partition table"
 printf ";\n;\n;\n;\ny\n" | sfdisk --DOS --force "${D}" > /dev/null 2>&1
done

```


```

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 is 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 141 or [Adding Custom Attributes to OS Installation Profile \(SA Client\)](#) on page 142.

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 12 Windows Custom Attributes for OS Provisioning

Keyword	Corresponding unattend.txt Attribute	Description
AdminPassword	[GuiUnattended]/ AdminPassword	This option sets the Administrator password for the Admin account.
argstring	None	String value that is used to compose the command line arguments for the 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 SA. This custom attribute should only be set on the Server, but SA does not prevent you from setting the attribute anywhere. The default value is an SA-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 Ghost™. However, using the built in support for WIM images is strongly encouraged.
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 SA.

Table 12 Windows Custom Attributes for OS Provisioning (cont'd)

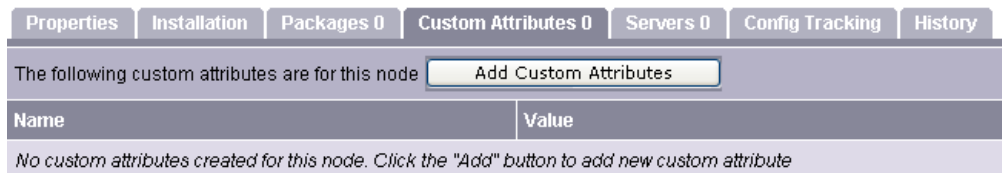
Keyword	Corresponding unattend.txt Attribute	Description
timeout	None	An integer value in minutes that the Windows Setup will timeout. Default is 120 minutes. If Windows setup does not complete in the specified amount of time, the OS installation fails with a timeout error.
DOSPartitionUtil	None	Used to pick one of the built-in utilities and sets of partitioning commands. Values can be: an be ms_fdisk, aefdisk, freedos_fdisk, or efdisk.
DOSPartitionOverride	None	The value should be the custom partitioning code used, not any of the built-in codes. For example: <pre>%tmp%aefdisk\aefdisk.exe 1 / delall /formatfat /pri:8000:0c / deactivate /mbr</pre>

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 45](#) shows.

Figure 45 The Custom Attributes Tab for an OS Installation Profile in the SAS Web Client




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 (SA Client)

To add custom attributes to an OS installation profile in the SA Client, perform the following steps:

- 1 Launch the SA 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** ► **SA 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)

SA 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.

Hardware Support in OS Provisioning

This section provides information on hardware support in OS provisioning within SA 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 SA. At a minimum, you must update the boot processes for Windows and Linux (SA 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 146 and [Adding Hardware Support to a Linux or VMware ESX Build Image](#) on page 150 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 SA was installed with the HP BSA 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 SA as the second stage PXE image for PXE network bootstrap programs such as PXELinux.

For Linux, SA 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. SA 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, SA 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. SA includes the Build Image Administrator, a tool for creating a boot floppy for Windows.

See [Creating a Linux or VMware ESX Boot Image](#) on page 150 for more information.

NIC Support in Solaris Boot Images

The SA Solaris SPARC and x86 boot images are based upon Solaris 10 U3 (11/06). They should provide the same hardware compatibility as Solaris 10 U3.

Please refer to the Sun hardware compatibility list for compatible hardware:

<http://www.sun.com/bigadmin/hcl/>

NIC Support in Red Hat Linux Boot Images

The SA Linux 4 boot image is based upon Red Hat Enterprise Linux 4 AS Update 4. It should provide the same hardware compatibility as RHEL 4. Please refer to the Red Hat hardware compatibility guide for a list of compatible hardware.

https://hardware.redhat.com/hwcert/list.cgi?component=Red%20Hat%20Enterprise%20Linux&version=4&bug_status=Compatible&showall=1

The SA Linux boot image is based upon Red Hat Enterprise Linux 3 AS Update 8. It should provide the same hardware compatibility as RHEL 3. Please refer to the Red Hat hardware compatibility guide for a list of compatible hardware.

https://hardware.redhat.com/hwcert/list.cgi?component=Red%20Hat%20Enterprise%20Linux&version=3&bug_status=Compatible&showall=1

NIC Support in Windows Boot Images

SA 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 SA 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 Build Image Administrator.

Each properly formatted cabinet file in the directory `\content\drivers\ndis` under the Build Image Administrator directory is included as an entry in the driver catalog.

Table 13 NIC Drivers Included with the Windows Boot Image

Driver Name	Description
B57	Broadcom NetXtreme Gigabit Ethernet NDIS2 Driver v8.28 (29 nov 05)
BXND20X	Broadcom NetXtreme II Ethernet
DC21X4	Digital 2104x/2114x 10/100 mbps Ethernet Controller v3.00
E1000	Intel 8254X Based Adapter (pro/1000 gigabit) v4.54 (06/28/05)

Table 13 NIC Drivers Included with the Windows Boot Image (cont'd)

Driver Name	Description
E100B	Intel PRO/100 Network Connection Driver v4.47 (03/05/03)
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 v1.9.005
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)

Table 14 NIC Drivers Included with the Windows WIN-BCOM DOS Boot Image

Driver Name	Description
B57	Broadcom NetXtreme Gigabit Ethernet NDIS2 Driver v10.03 (020107)
BXND20X	Broadcom NetXtreme II Ethernet
DC21X4	Digital 2104x/2114x 10/100 mbps Ethernet Controller v3.00
E1000	Intel 8254X Based Adapter (pro/1000 gigabit) v4.54 062805
E100B	Intel PRO/100 Network Connection Driver v4.47 030503
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 v1.9.005
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

Table 14 NIC Drivers Included with the Windows WIN-BCOM DOS Boot Image

Driver Name	Description
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 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 Phoneline 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 Build Image Administrator is used to create a Windows Boot Image that installs the OS Build Agent on servers. The Build Image Administrator is packaged with MSI.

You must meet the following requirements to use the Build Image Administrator:

- The machine on which the Build Image Administrator is installed must have a Python interpreter installed. You can obtain a Python interpreter from Active State Software: <http://www.activestate.com>.
- SA 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 146 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 Build Image Administrator by downloading the file `opswbia-<version>-0.msi` from the SAS Web Client.

Where `<version>` is the latest version of the Build Image Administrator tool for the release of SA installed at your facility. Only one version of the Build Image Administrator tool is available on the Software Repository.

- 2 Install the MSI package that contains the Build Image Administrator tool on a Windows server that has a Python 1.5.2 interpreter.

By default, the Build Image Administrator is installed in the following directory:

```
%SystemDrive%\Program Files\OPSWBIA
```

- 3 Change directories to the 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>
```

Options for the Build Image Administrator

You can use the options described in [Table 15](#) when you run the Build Image Administrator from the command line.

Table 15 Build Image Administrator Command Line Options

Option	Description
-a <drive>	Writes the boot image to this drive (Default drive: A).
-c	Makes an El Torito bootable CD image in addition to the boot image.
-d	Enables debugging of the OS Build Agent in the generated image.
-f	Formats the disk first when writing to a disk.
-h <host>	Specifies the host name for the Agent Gateway (required option).
-i <file>	Specifies the file name for the generated boot image (Default file name: dosopsw.1).
-n <directory>	Specifies the directory where the NDIS driver packages are located (Default directory: ./content/ndis).
-p <port>	Sets the port for the OS Build Agent to use to contact the Agent Gateway (Default port: 8017).
-t	Performs a test image generation and does not execute any commands.
-w	Writes the generated image to a disk in the drive specified by the option -a.

Updating PXE Image for Windows

When SA was installed with the 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 146.

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 Build Server.

Adding Hardware Support to a Linux or VMware ESX Build Image

You can modify SA 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, SA uses the two following types of Linux build images:

- **A Linux Boot Image:** SA 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:** SA 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.

After a fresh-install of an SA 7.80 Core, the Linux Build Images are located on the Build Manager host in the following directories:

```
/opt/opsware/boot/tftpboot/<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

SA 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 SA 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 and install it.
- 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>
```


4 Software Discovery

The HP Server Automation (SA) Software Discovery feature provides a signature-based software discovery mechanism for Windows and UNIX managed servers to help you manage applications and software that are not already managed by SA. Specifically, the Software Discovery feature:

- Discovers unlicensed software, unregistered software, custom-built software and software that was not installed using one of the standard packaging technologies for Windows or UNIX.
- Creates an inventory of software programs that have not been installed as an OS-registered application.
- Provides system administrators the ability to create snapshots of all the discovered software on a server and then periodically audit against the snapshot over time. This can help you upgrade a server, remove unwanted software and modify a server to conform to your organization's security policies.
- Gives auditors a convenient method of capturing the current state of a server's software and discovering unsupported or unlicensed software installed on a server. By running the audit on a regular schedule, you can monitor software changes over time.

Software Discovery is a read-only server module that provides a rich inventory of software on a managed server.

Software Discovery Prerequisites

To deploy the Software Discovery feature, you must meet the following requirements:

- SA 7.50 or later, installed and configured.
- SA patch 7.50.01 or later, installed on the SA 7.50 core.
- A BSA Essentials Network account. To request a BSA Essentials account, see <http://www.hp.com/go/bsanetwork>.
- The HP BSA connector (previously called the Live Network connector or LNC) installed and configured on your core server. See the *HP Live Network connector Installation and Configuration Guide* which is available from the BSA Essentials web site, <http://www.hp.com/go/bsanetwork>.
- If you want to use SAR (Service Automation Reporter) to create reports about discovered software, you must first subscribe to the SAR software_discovery_reports stream. See [Generating Reports with SAR](#) on page 154.



Software Discovery is not supported on IA64 Linux.

Generating Reports with SAR

If you want to use SAR (Service Automation Reporter) to create reports about discovered software, you must first subscribe to the SAR `software_discovery_reports` stream. If you do not download this stream and you use SAR, some of the discovered software data will not appear in your reports generated from SAR. If you do not plan to use SAR, you do not need to download this stream. Use the following command to display the SAR streams:

```
live-network-connector list-streams --product=sar
```

Set the `software_discovery_reports` line to 1 in the `live-network-connector.conf` file:

```
# SAR Software Discovery Reports stream
software_discovery_reports=1
```

For more information, see the *HP Live Network connector Installation and Configuration Guide* available on the BSA Essentials Network web site, <http://www.hp.com/go/bsanetwork>.

For information about SAR, see *HP Service Automation Reporter User Guide*.

Supported Platforms and Configurations

Software Discovery is supported on all UNIX platforms and Windows versions which the SA Agent supports for managed servers.

For more information on managed servers supported platforms, see the *SA Release Notes* or the *SA Planning and Installation Guide*.

Software discovery is ISO-8859-1 compliant.

Downloading the Software Discovery Package

The Software Discovery package is available from the BSA Essentials Network. To get the Software Discovery package, you must use the BSA connector (previously called the Live Network connector or LNC).



For more information on BSA Essentials and to request a BSA Essentials account, visit the BSA Essentials web site at <http://www.hp.com/go/bsanetwork>.

The BSA connector downloads the Software Discovery package in the form of a zip file with the following name:

```
OPSWsmo_discovered_software-<version>.zip
```

This zip file is placed in the following directory on your SA core server:

```
/var/opt/opsware/ogfs/mnt/root/var/opt/opsware/sm
```

This directory can also be accessed from the SA Global Shell at the following directory:

```
/var/opt/opsware/sm
```

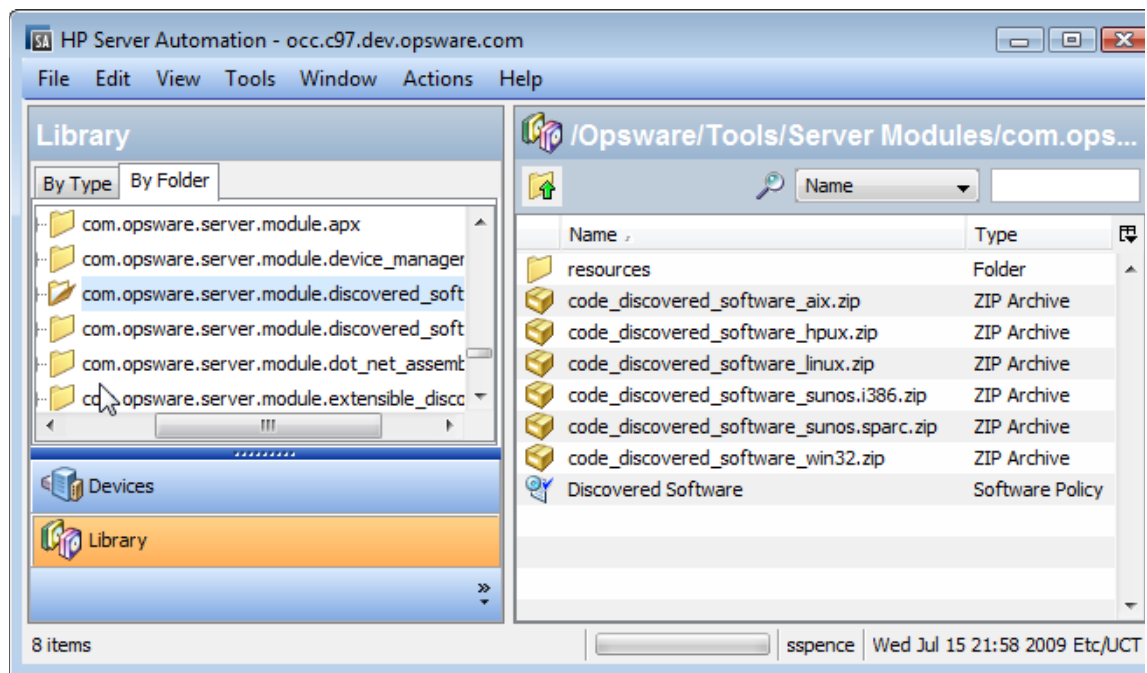
Contents of the Software Discovery Package

After you install the Software Discovery package, you can see the contents in the SA Client as follows.

- 1 In the SA Client, select Library in the navigation pane.
- 2 In the Library, select the By Folder tab.
- 3 Navigate to Library/Opware/Tools/Server Modules/com.opsware.server.module.discovered_software. This displays the following contents of the Discovered Software package as shown in Figure 46.
 - A set of zip file packages, one for each supported managed server platform.
 - A software policy named Discovered Software that contains all the zip file packages.
 - A subdirectory named resources.

The Discovered Software policy is automatically remediated (installed) when you initiate a snapshot with the “Perform Inventory” option selected, as described in [Deploying Software Discovery Using Inventory Snapshots](#) on page 156.

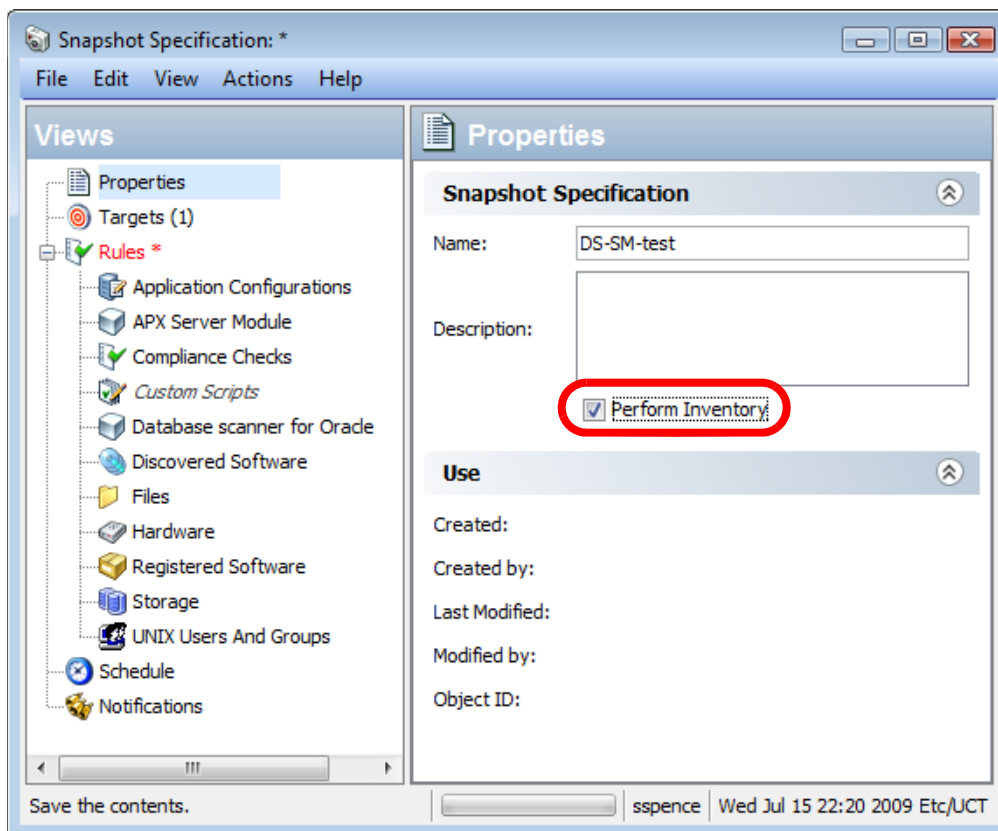
Figure 46 SA Client Library Showing Location of Software Discovery Packages



Deploying Software Discovery Using Inventory Snapshots

To deploy the Software Discovery feature to discover software installed on a server, you must run a snapshot with the “Perform Inventory” option selected, as shown in [Figure 47](#). For details, see “Creating a Snapshot Specification” and “Running a Snapshot Specification” in the *SA User Guide: Application Automation*.

Figure 47 Snapshot with Perform Inventory Option Selected to Enable Software Discovery



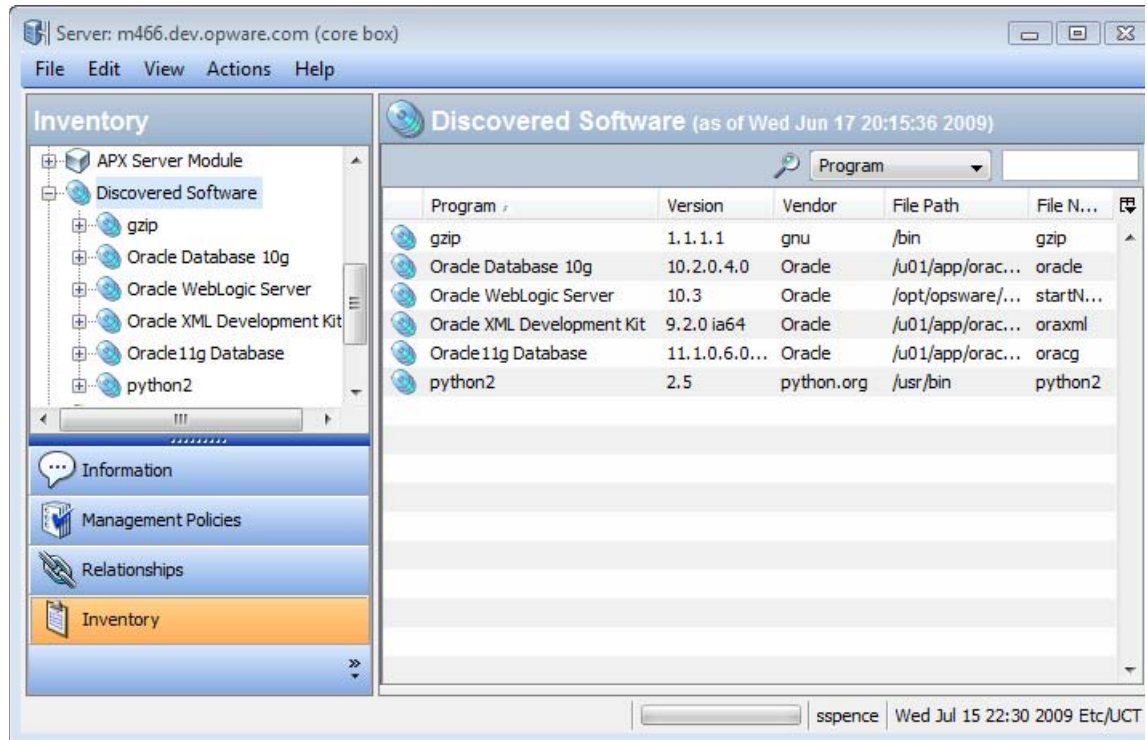
Each time a snapshot is run on a server with Perform Inventory selected, SA verifies that the Software Discovery server module is installed on the managed server. If the Software Discovery server module is not on the target server, a remediate job is launched automatically by the core that installs the contents of the Software Discovery Software Policy. The policy remediation occurs automatically when the snapshot is run.

This installs the Software Discovery server module on the server, including the signature zip package, which in turn enables you to view Software Discovery on a managed server’s Device Explorer, and use the Snapshot’s software inventory in an Audit.

Viewing Discovered Software on a Managed Server

The SA Client displays the software that has been discovered on the server in the Device Explorer in the Inventory view, as shown in [Figure 48](#).

Figure 48 Device Explorer Showing Software Discovered



Unlike other server objects in the managed server inventory that fetch data in real time from a server, the Discovered Software server object is configured to only show data from the latest inventory Snapshot (a Snapshot that has the Perform Inventory check box selected when the snapshot specification is created). If there are no inventory snapshots available, the server browser will show an error message suggesting that an inventory Snapshot should be made.

The Device Explorer displays a time stamp at the top of the Device Explorer window indicating the last time the inventory Snapshot was run on the server. For example, in [Figure 48](#), the Contents pane (right side) of the window shows the label: “(as of Thu May 29th 23:45:29 2008)”. This indicates the last time the inventory Snapshot was run.

For more information, see “Creating a Snapshot Specification” and “Running a Snapshot Specification” in the *SA User Guide: Application Automation*.

Inventory Snapshots in Audits

Once you have performed an inventory Snapshot, you can add it as the source of an audit, so whenever the audit runs, it will compare the original state of the snapshot with the current state of the server, and you can determine if the installed software has changed since you last ran the Snapshot.

For more information on Snapshots and Audits, see “Audit and Remediation” in the *SA User Guide: Application Automation*.

Inventory Snapshot Job Error Messages

The following error codes and messages are specific to Software Discovery when running an inventory Snapshot:

- * 1-TADNSW unexpectedly quit!

- * 2-Invalid JSON format
- * 3-Another instance of DS-SM is running.
- * 4-Database merge failed
- * 6-TADNSW run failure
- * 8-TADNSW Error: <error>

Software Discovery Permissions

The Software Discovery feature can be run on a server by any SA user with read access to the server and belongs to a user group that has the client feature permission “Allow Execute Server Modules: Yes”.

If a user wants to add or modify custom entries for Software Discovery, then the user will require the “Manage Server Modules: Read & Write” permission. Permissions are granted in the SAS Web Client.

For more information on SA permissions, see the *SA Administration Guide*.

How Software Discovery Works

Software Discovery provides a signature-based software discovery mechanism for SA Windows and UNIX managed servers. It consists of discovery logic, module metadata, and a signature database. This signature database contains application signatures from HP's Asset Management Tool, Discovery Dependency Mapping Inventory (DDMI), which is the Software Application Index (SAI) used by DDMI to discover software.



This feature is not designed to allow you to install or remediate software on a managed server. For information on using software policies to install and remediate software onto managed servers, consult the *SA User Guide: Server Automation*.

Application Discovery

Software Discovery uses DDMI SAI content for discovering applications on SA managed servers. The SAI signatures are used to generate the SCT (Signature Component Table) for use with Software Discovery. The SCT contains all the signatures imported from the DDMI SAI which are to be used by Software Discovery.

SCT application signatures are grouped together by product IDs and compared to the results retrieved by a file system scan. For every file found on the file system, a hash is computed using the DDMI hash algorithm. The signatures collected from the file scan are referred as 'raw' signatures. Application signatures represented in the DDMI SAI and corresponding SCT contain the same hash values computed from the same algorithm. The 'raw' signatures are then compared to the DDMI SAI signatures stored in SCT and a rating is calculated off the best possible match. The highest rated match is then reported as the application back to the user who initiated the scan.

All components are compared to the corresponding properties obtained from raw signatures. When a match occurs in any one of the components, the rating is incremented and used for gauging the best possible match among various product versions. When the best match is found, the product is reported using the display components:

```
DISPLAY PRODUCT: <display product-name>
DISPLAY VENDOR: <display vendor-name>
DISPLAY VERSION: <display version-number>
```

There are a few differences to note between the DDMI and Software Discovery results that are accepted around specific boundary conditions. The first difference in comparison is that Software Discovery will report all instances of software discovered on a server. For example, when evaluating multiple installs of the Java JRE, DDMI will only report the first install of a series of products of the same version. On the other hand, Software Discovery will report all instances of the installed product.

The second difference occurs when no exact matching signature is available in the SAI. In this situation, DDMI and the Software Discovery feature will attempt to find the next best match based off the ratings calculated. DDMI will evaluate all signatures, along with all 'associated' entries, to generate a high rating during guess estimates and find more accurate results. The Software Discovery feature will sometimes reach the same results in the guess. However, the guess does occasionally differ due to remaining 'associated' entries not being imported into SCT from the SAI.

Please note again that due to sizing constraints, not all SAI associated entries are imported into SCT. Only overlapping 'associated' entries between concurrent applications versions will be imported as well. As a result, SA will not pinpoint or discover suites or editions of applications.

Application Signatures

All application signatures are generated using the DDMI SAI Master.xml (WIN32) and unix.xml (UNIX). All application signatures are stored in a database for processing and identification during the scan process. SCT is used for the software discovery process in the database.

During a core install, these application signature packages must be uploaded to the core and attached to the Software Discovery Software Policy to which the Software Discovery package is attached. During upload, each package is associated with all UNIX or all Windows platforms. These packages are automatically installed onto a managed server when the server is remediated with respect to this policy. These signatures are used by the underlying scanning software to determine what software is discovered on a server.

Application Signatures from BSA Essentials Network

The Software Discovery feature comes with a database consisting of application signatures derived from the DDMI SAI content. When the Software Discovery package is imported into the SA core, all applications represented by those signatures are discovered. This signature database is updated periodically (usually monthly). These updates are provided through BSA Essentials Network and must be imported into the SA core.

The Software Discovery signature update package is a zip file, which has a file name similar to:

```
OPSWsmo_discovered_software_sig_prod-<version>.zip
```

After the BSA connector downloads the zip file, it is stored in the following directory on the SA core:

```
/var/opt/opsware/ogfs/mnt/root/var/opt/opsware/sm/data
```

This location can also be accessed from the Global Shell with the following command:

```
/var/opt/opsware/sm/data
```

The zip file can be uploaded to the core with the following dstool command:

```
/opt/opsware/smtool/dstool --user=<username> --pass=<password>  
OPSWsmo_discovered_software_sig_prod-<version>.zip
```

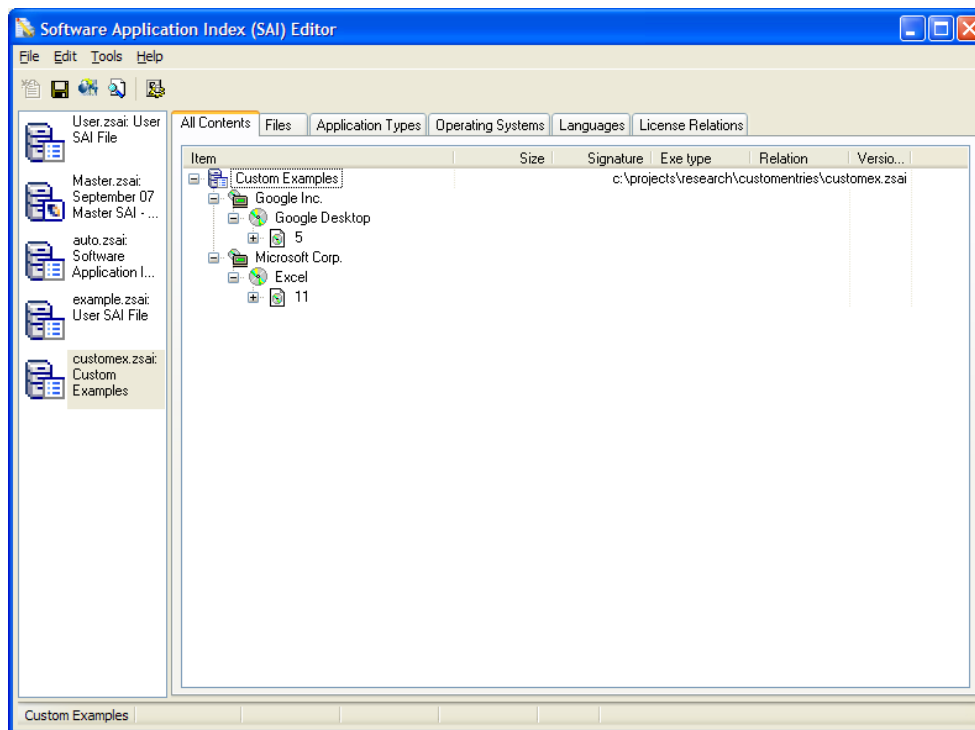
The --help option gives additional details on how to use dstool:

```
/opt/opsware/smtool/dstool --help
```

Custom Signatures

You can add custom application signatures using the DDMI SAI Editor and corresponding platform scanners as shown in [Figure 49](#).

Figure 49 Software Application Index (SAI) Editor



The SAI editor provides a convenient way to add new publishers and applications. Its scanners retrieve raw application signatures related to the specific applications you are interested in reporting as applications to SA.

It is recommended that you run the scanner prior to working with the SAI editor. The entire file system or specific directories can be targeted with the scanner by using the following command line options:

```
scanwin32-x86-2.50.000.7199.exe -fast -p:C:\projects\research\edscan\  
-paths:"C:\Program Files"
```

(scanwin32-x86-2.50.000.7199.exe is the latest version of the executable and may change without notice.)

In the SAI Editor, you can add the publisher, application, release, and version details for the specific applications that need to be categorized in SAI as shown in [Figure 50](#) through [Figure 53](#).

Figure 50 Publisher Properties

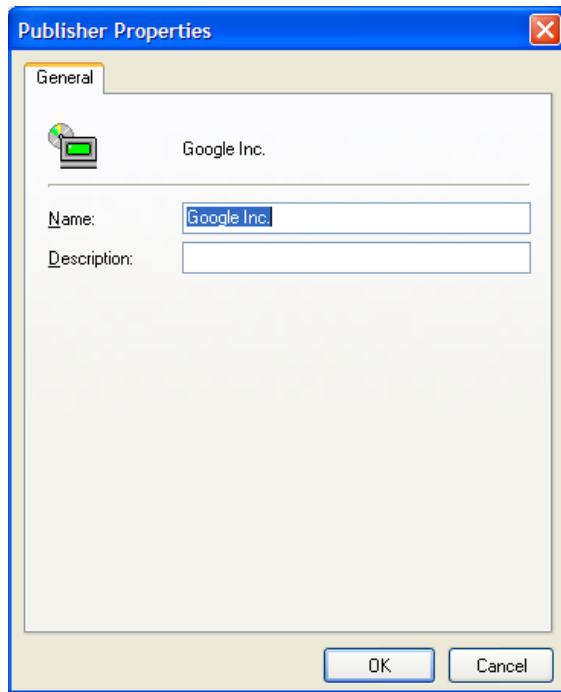


Figure 51 Application Properties

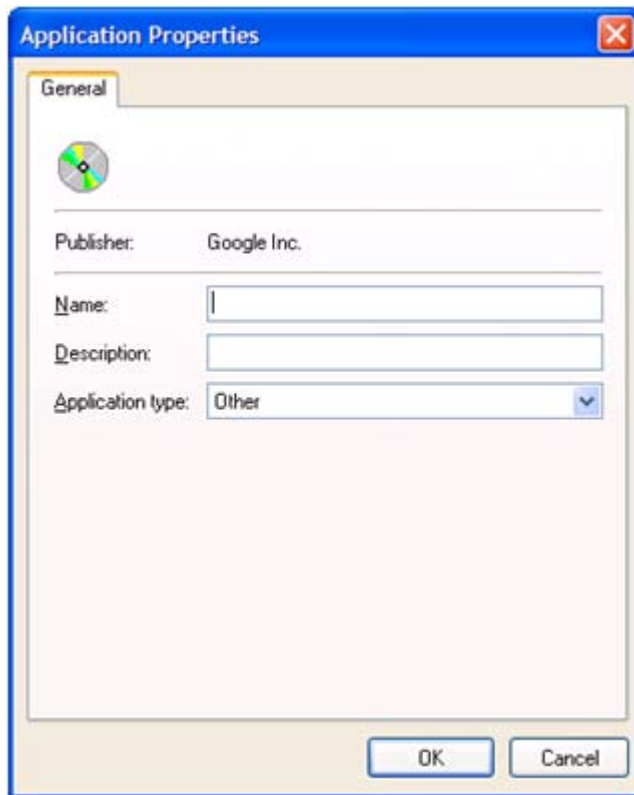


Figure 52 Release Properties

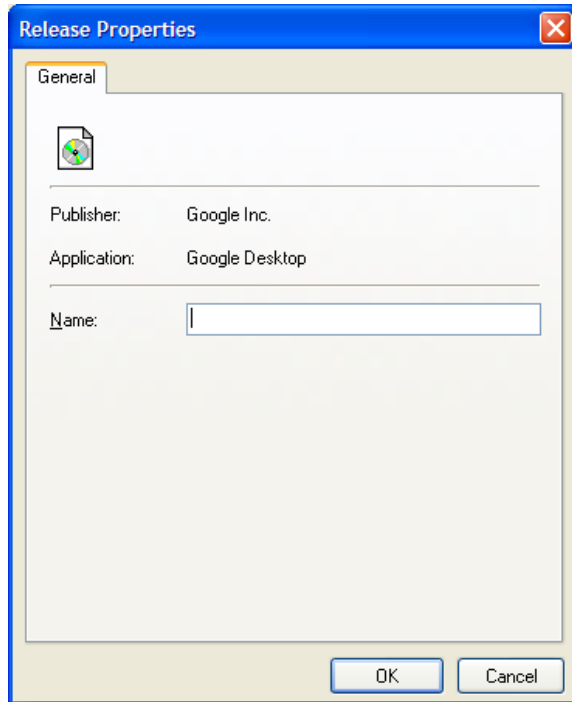
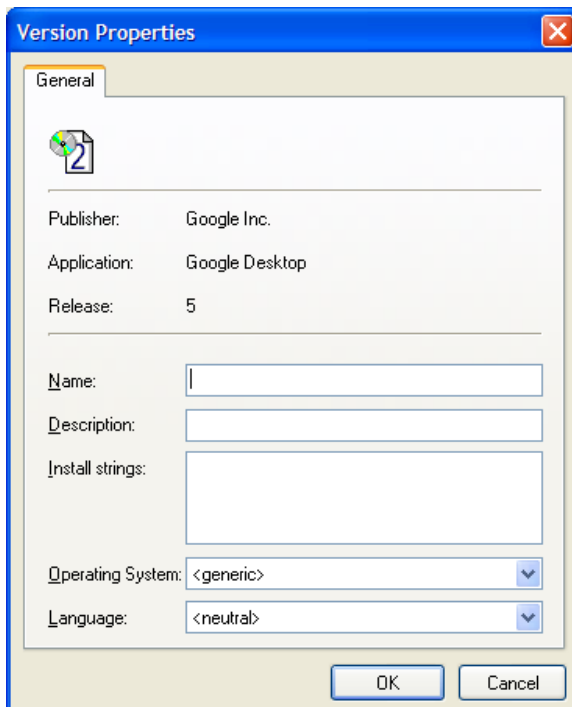


Figure 53 Version Properties



At this stage, you can open the scan results in the SAI Editor and begin adding corresponding signatures to the versions previously added in the SAI Editor as shown in [Figure 54](#) and [Figure 55](#).

Figure 54 Recognition Verification

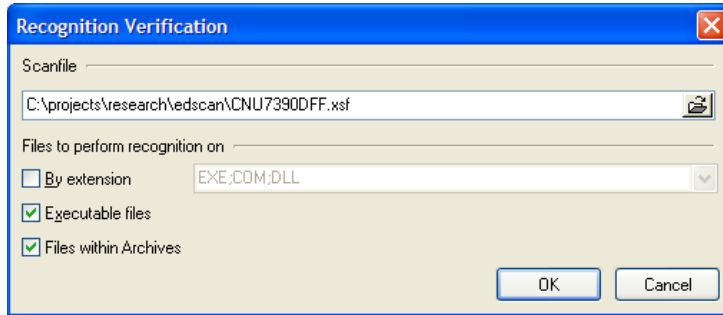
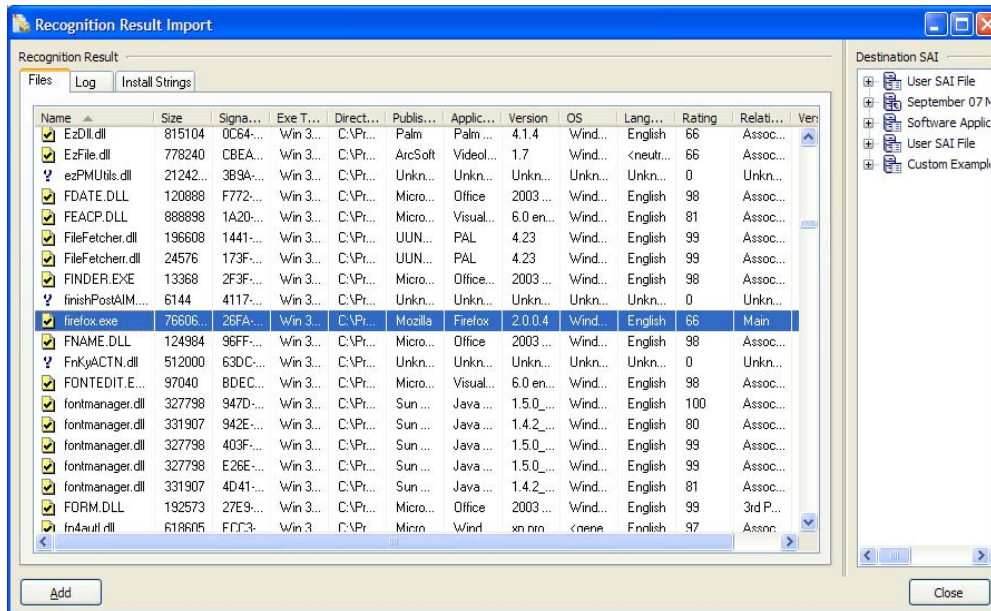
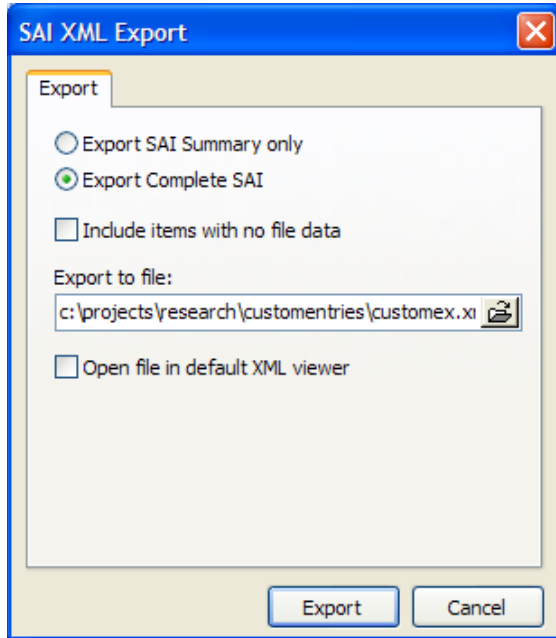


Figure 55 Recognition Result Import



Once the content is modified, you can export the content to the SAI XML database (User XML) as shown in [Figure 56](#).

Figure 56 SAI XML Export



The User XML database is then copied using the following command to the core where it is used by dstool:

```
/opt/opsware/smtool/dstool --username=<username> --password=<password>  
customex.xml
```

dstool generates a new (User) SCT database which is uploaded as a custom dependency to Software Discovery. During the next inventory snapshot, Software Discovery checks for the custom database and uses it along with the production database to categorize new applications.

If at any stage a mistake is made or you choose to remove all custom entries, invoke the dstool with the following command line options:

```
/opt/opsware/smtool/dstool --remove --username=<username>  
--password=<password>
```

Use the following --help option for additional details about the usage of dstool:

```
/opt/opsware/smtool/dstool --help
```

Please refer to the DDMI SAI Editor and scanner documentation for more details.

Configuration and Customization

UNIX and Windows Software Discovery support configuration attributes which drive how they run and the profile of resource usage on a managed server.

Configuration Attributes

The name of the custom attribute is HPSW_DS_SM_CONFIG.

The value of this custom attribute is in INI file format in Windows. Windows filters are grouped under the `FILTERS_WINDOWS` section while UNIX filters are grouped under the `FILTERS_UNIX` section. The `METHODS` section specify the scanning settings and the `CONFIGURATION` section contains other configuration settings.

The following is the syntax of a custom attribute:

```
[FILTERS_WINDOWS]
FILTER#={INCLUDE | EXCLUDE | EXIST}, { FILENAME | FILEPATH | FILESYSTYPE |
VOLUME | PRODUCT | VENDOR | VERSION}, criteria

[FILTERS_UNIX]
FILTER#={INCLUDE | EXCLUDE | EXIST}, { FILENAME | FILEPATH | FILESYSTYPE |
VOLUME | PRODUCT | VENDOR | VERSION}, criteria

[METHODS]
DELTA_SCAN= {TRUE | FALSE}

[CONFIGURATION]
LOG_MODE= {LOW | MEDIUM | HIGH | DEBUG}
INCLUDE_SYMBOLIC_LINKED_FILES={TRUE | FALSE}
INCLUDE_SYMBOLIC_LINKED_DIRS={TRUE | FALSE}
```

Configuration Options

Table 16 describes the configuration options used within the HPSW_DS_SM_CONFIG custom attribute.

Table 16 Configuration Options

Attribute	Platform	Description
'FILTER#'	All	<p>Create a list of filters for the Software Discovery Gauntlet for Cataloged and/or Uncataloged software. Filters are separated into two categories under INI format according to platform, [FILTERS_WINDOWS] and [FILTERS_UNIX]. Filter arguments are provided below their corresponding platform in the format of FILTER#=Action,Type,Criteria.</p> <p>Action can be one of 'Include', 'Exclude', or 'Exist'. Supported types are 'VOLUME', 'FILEPATH', 'FILENAME', 'PRODUCT', 'VENDOR', 'VERSION', 'FILESYSTYPE'. Criteria must correspond to the type.</p>
'INCLUDE_SYMBOLIC_LINKED_FILES'	UNIX	<p>True/False</p> <p>Default is False.</p> <p>Enable if you wish to include symbolic linked files in the scan.</p>
'INCLUDE_SYMBOLIC_LINKED_DIRS'	UNIX	<p>True/False</p> <p>Default is False.</p> <p>Enable if you wish for tadnsw to traverse into symbolic linked directories.</p>
'DELTA_SCAN'	All	<p>Enabled/Disabled or True/False.</p> <p>Default is False.</p> <p>Enable/Disable the delta scanning module.</p>
'LOG_MODE'	All	<p>Value is one of "LOW", "MEDIUM", "HIGH" or "DEBUG", default is "LOW".</p> <p>Sets the amount of information written to the log file. "LOW" is used to provide general runtime progress.</p> <p>"MEDIUM" will provide additional details regarding progress along with location if performing filescan.</p> <p>If "DEBUG" or "HIGH" is selected, the module will write enough information to the file so developers can diagnose obscure problems.</p>

Syntax Errors

Refer to [Inventory Snapshot Job Error Messages](#) on page 157 to see all the Software Discovery error messages and codes. Prior to using the syntax, the values in the custom attributes in [Table 16](#) are validated for proper syntax. If the configuration values violate the syntax, Software Discovery returns errors. Such messages are reported in place of `<error>` referenced in the list in [Inventory Snapshot Job Error Messages](#) on page 157. Validation only applies to the INCLUDE/EXCLUDE action assignments. If other invalid configuration options are specified, the default values override

Discovery Filtering Process

The use of Software Discovery filters benefits users by preventing unwanted data from consuming valuable network and database resources. The filtering scheme supports three actions, which are Include, Exist and Exclude.

The Software Discovery feature first checks the Include list to see what files should be examined. If a file is included, it progresses to the next step called Exist, which checks to make sure that selected fields contain certain data. Last, the discovery module compares the scanned data with all required fields against the Exclude list to see if the file should be excluded. If a file passes all configured filters, it is reported, otherwise the file is discarded and the discovery module continues scanning. All criteria fields accept inputs of `<criteria>`, prefix, postfix, as well as wild cards. If a wild card is specified at the end of the path string, the exact path is matched and sub-directories are not affected.

How Filters Affect Scanning

Include and exclude filters can affect the Software Discovery in a variety of ways. In general, the rules listed below apply.

Directory Scanning:

- If a directory is excluded, exclude all files and subdirectories.
- If a directory is included, include all files and subdirectories.
- If a directory does not match the include set, the status stays neutral and only the selected directories are scanned.
- If no directories are included or excluded, the status is set to include.

File Scanning:

- If a file is excluded, that file is not reported.
- If a file is included, that file is reported.
- If no files are included or excluded, the default status is set to include.

Sample Configuration

The following is a sample of a custom attribute in INI format.

```
[FILTERS_WINDOWS]
FILTER0=INCLUDE, VOLUME, "C:"
FILTER1=EXCLUDE, FILEPATH, "<WINDIR>\INSTALL*"
FILTER2=EXCLUDE, FILEPATH, "<WINSYSDIR>\DLLCACHE\"
```

```

FILTER3=EXCLUDE, FILEPATH, "<WINSYSDIR>\DRIVERS\"
FILTER4=EXCLUDE, FILENAME, "SETUP*"
FILTER5=EXCLUDE, FILEPATH, "\RECYCLE*"
FILTER6=EXCLUDE, FILEPATH, "\SYSTEM VOLUME INFORMATION*"
FILTER7=EXCLUDE, FILEPATH, "**\TEMPORARY INTERNET FILES\*"
FILTER8=EXCLUDE, FILEPATH, "**\LOCAL SETTINGS\TEMP\*"
FILTER9=EXCLUDE, FILEPATH, "**\LOCAL SETTINGS\HISTORY\*"
FILTER10=EXCLUDE, FILEPATH, "*DLLCACHE*"
FILTER11=EXCLUDE, FILEPATH, "*$NTUNINST*"
FILTER12=EXCLUDE, FILEPATH, "*$NTSERVICEPACKUNINSTALL$"
FILTER13=EXCLUDE, FILEPATH, "\I386"
FILTER14=EXCLUDE, FILEPATH, "*SP4\*"
FILTER15=EXCLUDE, FILEPATH, "*$HF_*"
FILTER16=EXCLUDE, FILEPATH, "*SERVICEPACKFILES*"
FILTER17=EXCLUDE, FILEPATH, "<WINSYSDIR>\WinSxS\"
FILTER18=INCLUDE, FILEPATH, "\DOCUMENTS AND SETTINGS*"
FILTER19=INCLUDE, FILEPATH, "\PROGRAM FILES*"
FILTER20=EXIST, PRODUCT
FILTER21=EXIST, VERSION

```

[FILTERS_UNIX]

```

FILTER0=EXCLUDE, FILENAME, "*.dll"
FILTER1=EXCLUDE, FILENAME, "*.com"
FILTER2=EXCLUDE, FILENAME, "*.cmd"
FILTER3=EXCLUDE, FILENAME, "*.html"
FILTER4=INCLUDE, FILEPATH, "/etc*"
FILTER5=INCLUDE, FILEPATH, "/opt*"
FILTER6=INCLUDE, FILEPATH, "/bin*"
FILTER7=INCLUDE, FILEPATH, "/usr/bin*"
FILTER8=INCLUDE, FILEPATH, "/usr/lib*"
FILTER9=INCLUDE, FILEPATH, "/usr/games*"
FILTER10=INCLUDE, FILEPATH, "/usr/sbin*"
FILTER11=EXCLUDE, FILEPATH, "/cygdrive*"
FILTER12=EXCLUDE, FILEPATH, "/lost+found*"
FILTER13=EXCLUDE, FILEPATH, "/proc*"
FILTER14=EXCLUDE, FILEPATH, "/tmp*"

```

[METHODS]

```

DELTA_SCAN=TRUE

```

[CONFIGURATION]

```

LOG_MODE=LOW
INCLUDE_SYMBOLIC_LINKED_FILES=FALSE
INCLUDE_SYMBOLIC_LINKED_DIRS=FALSE

```

Concurrency and Multi-User Considerations

Software Discovery is configured to look at a Snapshot before attempting to run the inventory Snapshot. If an inventory Snapshot is available, the results are immediately displayed instead of running a new inventory Snapshot. However, if two users initially run an inventory

Snapshot, one of the instances will be accepted and run an inventory Snapshot. The other user will see a message when attempting to view the Discovered Software server module in the Device Explorer alerting them that an inventory Snapshot is already in progress.

If Software Discovery is used during an ad-hoc or scheduled Snapshot job, the data from the Discovered Software server module is persisted as a Snapshot package (snapshot.zip) on the SA core. These zip packages are used, for example, by the SAS Web Client when it needs to perform an Audit operation.

Software Discovery Usage Examples

The Software Discovery discovery mechanism can be useful in case of inherited servers and other usage examples. Some examples are listed below:

Example 1

You are an IT Administrator at a company that has 500 managed servers in your data center. You want to know how many servers have 1.3 Java JRE installed. You create a new Audit and select a source server that you know has Java JRE installed. You perform the following steps:

- 1 From the Discovered Software tab under Rules select and expand Software Node.
- 2 Selects JRE installed on the source server and add an additional regex property check for version='1\3.*'.
- 3 Form the Targets tab select all the active servers in the lab.
- 4 Save and run the audit to check all the managed servers in the lab for 1.3 versions of Java JRE.

The results come back and you see there are still some servers that are not compliant and proceed to update the remaining servers.

Example 2

For the next task, you want to check all 64-bit machines and see what 32-bit applications were installed. You create a new policy and add your own configuration for Software Discovery using the HPSW_DS_SM_CONFIG custom attribute. In that attribute, you assign the following options:

```
[FILTERS_WINDOWS]
FILTER0=INCLUDE, VOLUME, "C:"
FILTER1=INCLUDE, FILEPATH, "\PROGRAM FILES (X86)*"

[METHODS]
DELTA_SCAN=TRUE

[CONFIGURATION]
LOG_MODE=LOW
```

You attach all the 64-bit Windows 2003 Servers to the new policy and remediate the machines. Next, you create a new snapshot including all 64-bit Windows 2003 Servers and select the wildcard under the Discovered Software tab. You save the snapshot and proceed to

run the snapshot against all the 64-bit Windows 2003 Servers. Results come back and now you have a list of all the 32-bit applications deployed across all the 64-bit Windows 2003 Servers.

Example 3

You are auditing the lab to make sure all Windows servers have the appropriate virus detection software installed. You create a new Audit and select a source server you know has the correct software installed. You navigate to the Discovered Software tab and select Norton Antivirus under the Software node.

You include all the Windows machines in the lab as target machines and then save the Audit. The Audit is then run and the results come back with half the machines in the lab non-compliant. You proceed to install Norton Antivirus on the remainder noncompliant servers.

Extending Software Discovery

While Software Discovery can discover a very large number and a wide variety of applications, you can extend this feature by adding your own Python scripts to perform custom software discovery. To extend Software Discovery, perform the following steps.

- 1 Write your custom software discovery code in Python as described in [Writing Custom Software Discovery Code](#) on page 171.
- 2 Package your Python file into a zip file.
- 3 In the SA Client, select Library in the navigation panel.
- 4 Select the By Folder tab.
- 5 Create the following folder. You will import your zip file and create a software policy in this folder. Make sure you have adequate permission to create this folder. See the *SA Administration Guide* for details on permissions.

```
Library/Opware/Tools/Server Modules/  
com.opware.server.module.discovered_software.ext/
```

To create this folder, you need write permission on the parent folder. For more information, see “Folder Permissions” in the *SA Administration Guide*.

- 6 Import your zip file to this folder. Right click and select **Import Software...** or select the **Actions** ► **Import Software...** menu item. For details, see [Importing Packages](#) on page 69.
- 7 Open the zip package and set the Default Install Path to one of the following.

— For UNIX servers set the default install path to:

```
/opt/opware/sm/com.opware.server.module.discovered_software.ext
```

— For Windows servers set the default install path to:

```
%PROGRAMFILES%\Opware\sm\com.opware.server.module.discovered_software.ext
```

For details, see [Viewing Package Properties](#) on page 73 and [Editing Package Properties](#) on page 76.

- 8 In the same folder (from [step 5](#) above), create a software policy named `com.opsware.server.module.discovered_software.ext`. Right click and select **New Software Policy...** or select the **Actions ► New Software Policy...** menu item. For details, see [Creating Software Policies and Software Templates](#) on page 39.
- 9 Add your imported zip file to the software policy. Open the software policy, select the Policy Items view, and select the “+” button. For details, see [Adding Software Resources to a Software Policy](#) on page 44.
- 10 On the core server, run the `smtool` command with the `--upgrade` and `--force_upgrade` options. Specify the user name and password of an SA user with the “Manage Server Module: Read & Write” permission. Specify the Software Discovery zip package that you downloaded from BSA Essentials as described in [Downloading the Software Discovery Package](#) on page 154. For example:


```
smtool --username=<user name> --password=<password> --upgrade
--force-upgrade OPSWsmo_discovered_software-<version>.zip
```

This command adds your custom software discovery package to the Discovered Software policy. `OPSWsmo_discovered_software-<version>.zip` is the Software Discovery package downloaded from the BSA Essentials Network as described in [Downloading the Software Discovery Package](#) on page 154.

For more information on the `smtool`, run `smtool --help`. The `smtool` command is located on the SA core server in the directory `/opt/opsware/smtool`. For more information on SA permissions, see [Software Discovery Permissions](#) on page 158 and the *SA Administration Guide*.
- 11 Run your custom software discovery code as described in [Deploying Software Discovery Using Inventory Snapshots](#) on page 156.

Writing Custom Software Discovery Code

This section describes how to write custom Python code to discover software that is not already discoverable by the SA Software Discovery feature. For instructions on how to integrate your custom code into SA, see [Extending Software Discovery](#) on page 170.

- The Python script must contain a `discover()` function that takes a list argument named `objects`, and returns a tuple of `(objects, merge_flag)` where `objects` is a list and `merge_flag` is an integer.
- The `objects` argument contains all the software objects Software Discovery has found. The returned list contains either the list of objects passed into the `discover()` function merged with the software objects discovered by your extension, or only the software objects discovered by your extension, depending on the integer value returned in the second parameter. See below for an example.

- The integer value, `merge_flag`, returned by the `discover()` function must be 0, 1 or 2 as described in the following table.

Table 17 Integer Value Returned by the `discover()` Function

Value Returned	Meaning
0	Your extension has merged the software objects passed in to the <code>discover()</code> function with the software objects it has found and returns the union of these two sets.
1	Your extension returns only the software objects it has found. The Software Discovery server module will merge the returned objects with its list of software objects. If there is a conflict, the software objects discovered by your extension will take precedence.
2	Your extension returns only the software objects it has found. The Software Discovery server module will merge the returned objects with its list of software objects. If there is a conflict, the software objects discovered by the Software Discovery server module will take precedence.

- The file name of your Python script must be in the following format:

`ds_ext_<unique name>.py`.

The `<unique name>` must be unique in all extensions in both the HP Software Discovery directory and the software discovery extensions directory. As a best practice, use your company name in the `<unique name>` to avoid conflicts. For example, the following could be a script from the “abc company”: `ds_ext_abcco_app.py`.

- The `discover()` function will be called after the Software Discovery server module runs and before it returns the result to the caller.
- Below is a sample implementation of the `discover()` function that returns a list containing one simulated software object:

```
def discover(objects):
    obj = {
        'opswType': 'software',
        'FILENAME': '/usr/local/bin/',
        'FILEPATH': 'my_util',
        'primaryKey': '/usr/local/bin/my_util',
        'VERSION': '1.3.0.1',
        'DISPLAY_VERSION': '1.3',
        'PRODUCT': 'My Util',
        'DISPLAY_PRODUCT': 'My Util',
        'VENDOR': 'abc co',
        'DISPLAY_VENDOR': 'abc co',
        'VOLUME': '',
        'FILESIZE': 32656,
        'FILEVER': '1.3.0.1',
        'FILEDATE': '2008-12-02 01:03:38'
    }
    return ([obj], 1)
```

- After writing your software discovery code, integrate it into SA as described in [Extending Software Discovery](#) on page 170.

5 Code Deployment Setup

-
- ▶ You must have specific permissions to setup code and content by using the SAS Web Client. Contact your SA administrator to obtain the necessary access rights.
 - ▶ The Code Deployment and Rollback (CDR) and Configuration Tracking features are being deprecated in this release and will no longer be available as of SA 8.0. For more information, talk to your HP sales representative.
-

Code Deployment Process

This section provides information on the code deployment process within HP Server Automation and contains the following topics:

- [Deploying Code](#)
- [Uploading Code and Content to Staging](#)
- [CDR Operations and Directories](#)
- [CDR Features](#)
- [CDR Permissions](#)
- [Accessing CDR](#)

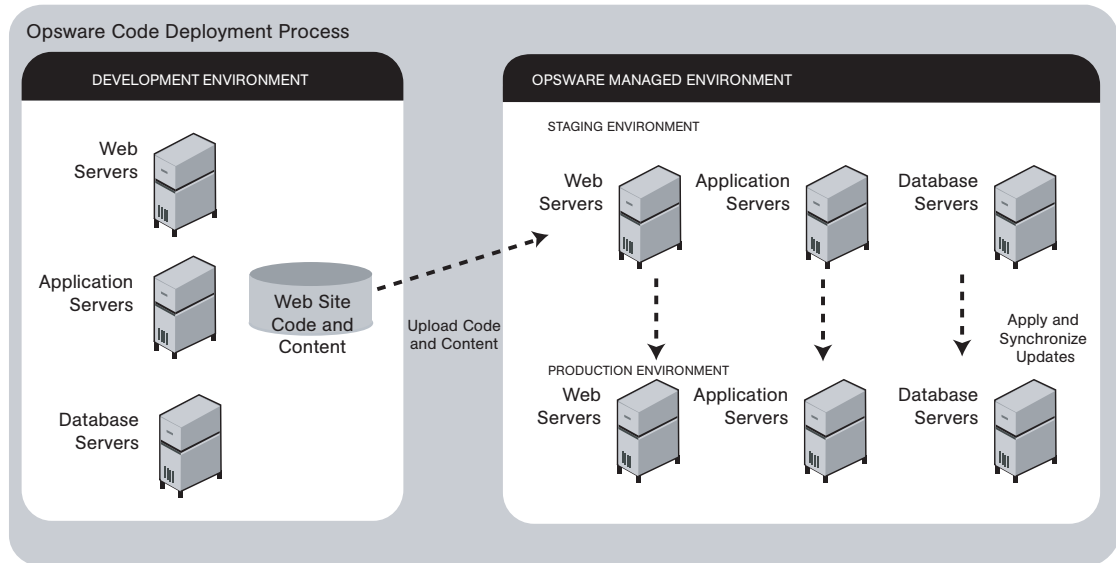
-
- ▶ The Code Deployment and Rollback feature is not supported on VMware ESX 3.0.
-

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 a managed environment.

Figure 57 Typical Code and Content Update in the SA 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 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 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.



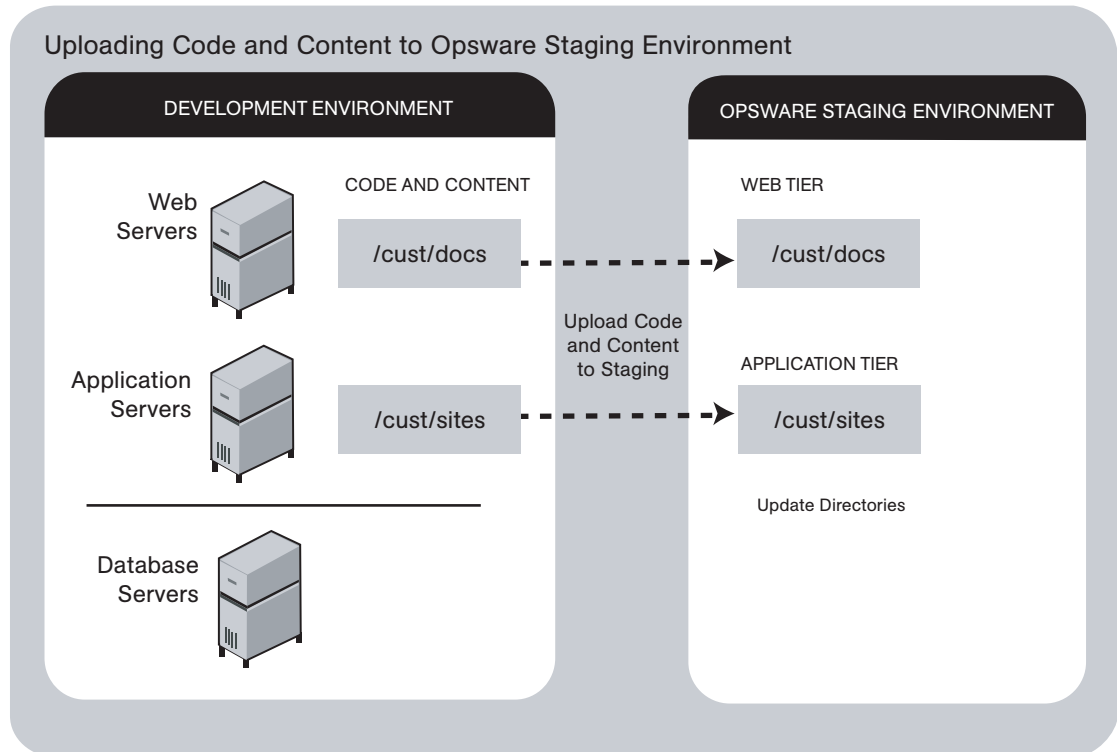
The code and content deployment process that you follow might be different depending on the architecture of your operational environment and your deployment requirements.

Uploading Code and Content to Staging

Before you use CDR to push code and content, you must upload new or updated files to your 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 58 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 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 HP Server Automation, 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 59 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 Defintions. 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 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 *SA 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 HP-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 a 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 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 SA 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 182 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 SA 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 189 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 *SA 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.

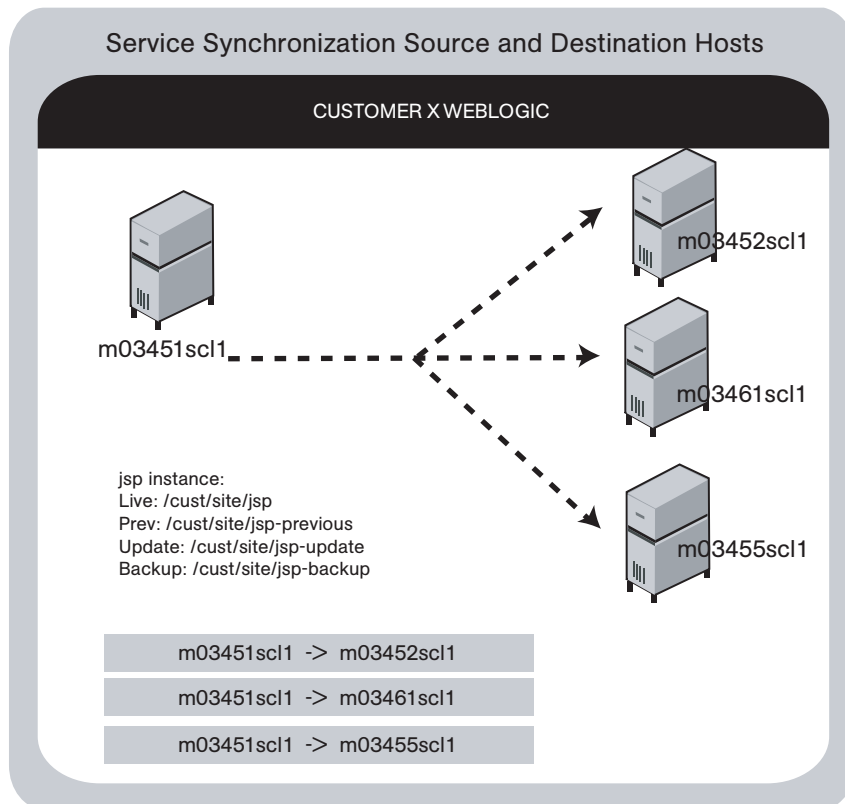
Planning Your CDR Configuration

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 60 shows an example of a typical synchronization diagram defined for a specific service, in this case, a WebLogic (jsp) application instance.

Figure 60 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 18 shows the information that you can specify for a service defined in CDR.

Table 18 Table of Information to Specify for Code Deployment Service

Section of Page	Field Name
Service	
	Name (required)
	Type (required)

Table 18 Table of Information to Specify for Code Deployment Service (cont'd)

Section of Page	Field Name
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 18 Table of Information to Specify for Code Deployment Service (cont'd)

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 19 shows information that you can specify for synchronizations defined in CDR.

Table 19 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

- 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 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 SA administrator defines in the Administration section of the navigation panel. [Table 20](#), [Table 21](#), [Table 22](#), and [Table 23](#) provide descriptions of permissions that are associated with predefined CDR user groups.

Table 20 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 21 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 22 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 23 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 *SA 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 SA administrator can add individual users to each CDR user group to control their permissions to request or perform service operations, synchronizations, and sequences.

See *SA Administration Guide* for information on how to add users to CDR user groups.

When a user selects a CDR option, HP Server Automation 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 182 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 HP Server Automation 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 61 shows.

Figure 61 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 62 shows. The drop-down list includes the names of all application instances defined in the Model Repository.

Figure 62 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 63](#)), 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 63 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 64](#).

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 64 CDR Service Hosts

Service Host(s)	
Hosts	<input type="text" value="m0178whitesox.cust.custqa4.com"/> <input type="text" value="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 65](#).

See [Access Control for CDR](#) on page 187 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 65 CDR Roles for Performers and Requesters

Roles	
Perform Role Name	<input type="text" value="Select a role"/>
Request Role Name	<input type="text" value="Select a role"/>

Figure 66 Email Addresses to Copy CDR Operation Requests To

Service Options	
CC Operation Requests To	<input type="text"/> (xxx@xxx.com,yyy@xxx.com ...)

- 10 In the Service Options section (see [Figure 66](#)), 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 67](#).


Figure 67 Define a New Synchronization Page

<input type="button" value="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"/>
<input type="button" value="Save"/> <input type="button" value="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 187 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.



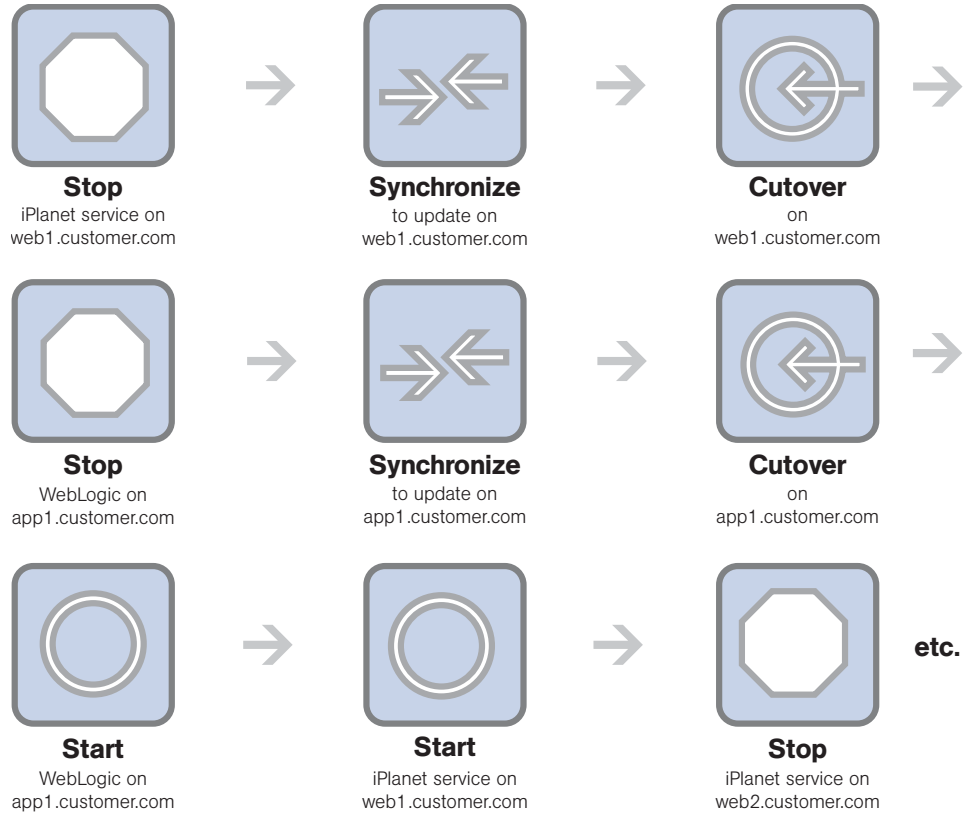
Deleting synchronizations that are used by a sequence causes that sequence to be deleted.

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 68 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 69](#).

Figure 69 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" value="(xxx@xxx.com,yyy@xxx.com ...)"/>
Email When Sequence Completes	<input type="text" value="(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 187 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 popsup 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 70](#).



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 70 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"/>

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 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.

Index

A

- accessing
 - CDR, 176
- adding
 - hardware support to Linux build images, 150
 - NIC support to Windows floppy images, 146
- address ranges, changing in IP ranges, 26
- administrators
 - creating, customers, 14
 - deleting, customers, 18
 - restrictions for deleting customers, 18
 - setting, custom attributes, 17
 - updating, customer information, 16

B

- boot floppies
 - SA Build Image Administrator options, 149
 - Windows servers
 - creating for, 148
 - overview, 143
- build customization scripts
 - Linux, overview, 129
 - overview, 122
 - requirements
 - for Linux, 129
 - for Solaris, 126
 - Solaris
 - overview, 125
 - sample, 127
- build images, adding hardware support for Linux, 150

C

- CDR. *See* code deployment.
- CIDR, changing in IP ranges, 26
- code deployment
 - access control, overview for setting up, 187
 - accessing, 176
 - code and content, uploading to staging, 175

- configuration
 - checklist, 180
 - planning, 182
 - procedures, 180
 - steps, 181
 - troubleshooting, 201
 - creating, directories on hosts, 186
 - determining, deployment requirements, 182
 - directories, populating initial content, 186
 - features, 176
 - hosts, preparing for CDR, 186
 - pre- and post-synchronization scripts, details of running, 193
 - sequences
 - creating, 198
 - deleting, 201
 - modifying, 200
 - services
 - creating, 189
 - modifying, 193
 - synchronizations
 - defining, 195
 - deleting, 197
 - modifying, 196
- Code Deployment and Rollback (CDR). *See* code deployment.
- configuration
 - checklist for CDR, 180
 - Code Deployment and Rollback feature, 181
 - planning for CDR, 182
 - procedures for CDR, 180
 - troubleshooting for CDR, 201
 - creating
 - CDR directories, 186
 - CDR sequences, overview, 197
 - CDR synchronizations, overview, 194
 - customer accounts, 14
 - deployment stage values, 22
 - directories on hosts for code deployment, 186
 - IP range groups, 25
 - Linux boot image, 150
 - media resource locators (MRLs), 101
 - sequences for CDR, 198
 - server use values, 20
 - services for CDR, 189

- synchronizations for CDR, 195
- Windows boot floppies, 148

custom attributes

- Linux OS provisioning, setting for, 137
- Solaris OS provisioning, setting for, 136
- Windows OS provisioning, setting for, 139

customer accounts

- creating, 14
- deleting, customers, 18
- setting, custom attributes, 17
- updating, 16

customers

- association with servers, 12
- Customer Independent, definition of, 11
- Not Assigned customer, definition of, 12

D

defining, synchronizations for CDR, 195

deleting

- customers, 18
- deployment stage values, 23
- media resource locators (MRLs), 102
- OS installation profiles, 121
- sequences for CDR, 201
- server use values, 21
- services for CDR, 194
- synchronizations for CDR, 197

deployment. *See* code deployment.

DHCP

- Linux servers, requirements for using, 150
- OS provisioning, usage of, 90
- Solaris servers, booting with, 123

directories

- code deployment, creating for, 186
- populating initial content for CDR, 186

E

editing

- deployment stage values, 23
- media resource locators (MRLs), 101
- server use values, 21

examples

- response file
 - for Windows 2000, 106
 - for Windows NT, 107
- sample mapfile for Intel Ethernet Adapter, 147
- sample Solaris build customization script, 127
- Windows sample mapfile, 147

F

firewall configuration, 104

firewall configuration for OS Provisioning, 104

Firewalls, 104

floppy images, prerequisites for Windows, 148

H

hardware support

- adding to Linux build images, 150
- OS provisioning, 143

histories

- viewing, changes in OS installation profiles, 120

I

Import Media Tool, creating MRLs, 101

IP addresses

- changing, CIDR in IP ranges, 26
- prefix length, decreasing in IP ranges, 27
- ranges, changing, 26
- status in IP range, changing, 28

IP range groups

- creating, 25
- overview, 24

IP ranges

- changing address ranges for, 26
- CIDR, changing, 26
- decreasing, prefix length, 27
- IP address status, changing, 28
- overview, 24

L

Linux

- build customization scripts
 - overview, 129
 - requirements for, 129
- creating, boot image, 150
- hardware support, adding to build images, 150
- PXE, using for booting servers, 143
- setting, custom attributes for servers, 137

M

mapfiles

- sample for Intel Ethernet Adapter, 147
- sample for Windows servers, 147

media resource locators (MRLs)

- creating, 101
- creating, prerequisites for, 94
- deleting, 102
- editing, 101

modifying

- sequences for CDR, 200
- sequences for CDR, overview, 197

- services for CDR, 193
- services for CDR, overview, 189
- synchronizations for CDR, 196
- synchronizations for CDR, overview, 194

N

- NIC support, Windows servers, adding, 146

O

- operating systems
 - defining for OS provisioning, 108, 112
- OS build process
 - default values for, 136
 - Solaris servers, 123
- OS installation profiles
 - deleting, 121
 - histories, viewing, 120
 - modifying, 117
 - modifying packages in, 119
 - overview, 103
 - properties, changing, 116
 - software, specifying, 104
 - working with, 108
- OS media
 - management, overview, 93
 - media resource locators (MRLs), creating, 101
 - prerequisites for creating MRLs, 94
- OS Provisioning, 104
- OS provisioning
 - hardware support, 143
 - Linux
 - custom attributes, setting up, 137
 - modifying operating system installation, 117
 - OS installation profiles, preparing, 108, 112
 - Prepare Operating System Wizard, 108, 112
 - setup
 - overview, 89
 - Solaris custom attributes, setting up, 136
 - Windows custom attributes, setting up, 139

P

- packages
 - modifying in OS installation profiles, 119
- permissions
 - code deployment, required for, 187
 - scripts, 55
 - sequence role for CDR, defined, 187
 - service role for CDR, defined, 187
 - special deployment role for CDR, defined, 187
 - synchronization role for CDR, defined, 187
- prefix length

- decreasing for IP ranges, 27
- increasing for IP ranges, 27

- Prepare Operating System Wizard, 108, 112

- prerequisites
 - MRLs, creating, 94
 - Windows floppy images, creating, 148
- properties, OS installation profiles, changing for, 116
- PXE images
 - overview for Windows and Linux, 143
 - Windows, modifying for, 149

R

- Red Hat Linux, 104

- response files
 - example
 - for Windows 2000, 106
 - for Windows NT, 107

- roles. *See* user roles.

S

SA

- Code Deployment & Rollback, features, 176
- server attributes, 19

- SA Build Image Administrator, options for, 149

- scripts, 55
 - Linux build customization scripts, 129
 - Linux servers, customizing build, 129
 - pre- and post-synchronization scripts for CDR, running, 193
 - Solaris build customization scripts, requirements for, 126
 - Solaris servers, customizing build, 125

- sequences
 - creating for CDR, 198
 - deleting for CDR, 201
 - modifying for CDR, 200

- server attributes
 - creating, deployment stage values, 22
 - creating, server use values, 20
 - deleting, deployment stage values, 23
 - deleting, server use values, 21
 - editing, deployment stage values, 23
 - editing, server use values, 21
 - overview, 19

- servers
 - association with customers, 12
 - preparing for code deployment with CDR, 186

- services
 - creating for CDR, 189
 - deleting for CDR, 194

- modifying for CDR, 193
- setting
 - custom attributes, 17
- setup for servers
 - Linux OS provisioning, 91
 - operating systems for provisioning, 108, 112
 - overview for OS provisioning, 89
 - Solaris OS provisioning, 90
 - Windows OS provisioning, 92
- software
 - specifying in OS installation profiles, 104
- Software Policies, Creating, 39
- Software Templates, Creating, 39
- Solaris
 - build customization scripts
 - overview, 125
 - sample, 127
 - custom attributes, setting for Solaris servers, 136
 - requirements for build customization scripts, 126
- synchronizations
 - creating for CDR, 195
 - deleting for CDR, 197
 - modifying for CDR, 196

T

- troubleshooting, CDR configuration, 201

U

- updating, customer accounts, 16
- uploading
 - code and content to staging for CDR, 175
- user roles
 - sequence role for CDR, 187
 - service role for CDR, 187
 - special deployment role for CDR, 187
 - synchronization role for CDR, 187

V

- viewing
 - CDR configuration, 201
 - changes for OS installation profiles, 120

W

- Windows floppy images
 - NIC support, adding, 146
- Windows servers

- boot floppies
 - creating, 148
 - overview, 143
- floppy images, prerequisites for creating, 148
- PXE, using for booting, 143
- PXE images, modifying, 149
- sample mapfile to build servers, 147
- sample response file
 - for Windows 2000, 106
 - for Windows NT, 107
- setting custom attributes for, 139

wizards

- Prepare Operating System, 108, 112