

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: 9.10

User Guide: OS Provisioning

Document Release Date: June 2011

Software Release Date: June 2011



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 2000-2011 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

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

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

Oracle and Java are registered trademarks of Oracle 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, 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	OS Provisioning Concepts	11
	The OS Provisioning Process	12
	SA OS Provisioning Components	15
	The OS Build Agent	15
	OGFS Agent	16
	The Build Manager	16
	The Media Server	16
	The Boot Server	16
	Build Customization Scripts	16
	How the OS Build Agent Locates the Build Manager	17
	WinPE	17
	Linux:	17
	Linux IA64:	17
	Solaris	17
	Non-DHCP Environments	17
2	OS Provisioning Requirements	19
	SA Core Setup Requirements	19
	Permissions	19
	Network Setup for OS Provisioning	19
	Firewall Considerations	20
	Windows Server 2008, Windows Server 2003 SP1 and Windows XP SP2 Firewall Settings	20
	Red Hat Enterprise Linux 4, 5 and 6 Firewall Settings	20
	Suse Linux Enterprise Server Firewall Settings	21
	Requirements for Provisioned Servers	21
	Supported Operating Systems and Media for OS Provisioning	21
	Supported Boot Media	22
	SA OS Provisioning-supplied CD Boot Images	22
	Itanium-Based Systems	22
	Solaris Servers	22
	SPARC SUN4U Servers	23
	Windows Servers	23
	Virtualization Support — VMware ESX, ESXi and Solaris 10	23
	HP-UX Servers	24
	OS Build Plan Requirements	24
	Hardware Preparation	25
	Windows Hardware Preparation Requirements	25
	Sun Solaris Hardware Preparation Requirements	26
	VMware ESXi Hardware Preparation Requirements	26

Red Hat Linux Hardware Preparation Requirements	26
Hardware Support in OS Provisioning	26
NIC Support in PXE Images for Linux	26
NIC Support in Solaris Boot Images	27
NIC Support in Windows Boot Images	27
Adding Hardware Support to a Linux or VMware ESX Build Image	28
3 Operating System Provisioning Setup	31
OS Provisioning Setup Task Summary	31
Setting Up the Media Server.	33
Creating Media Resource Locators (MRLs).	33
Import Media Tool Prerequisites	34
Import Media Tool Syntax and Options	34
Configuring the Media Server for Microsoft Windows OS Media/Image	36
Importing Windows Media from Linux Host	36
Importing Windows Media from a Solaris Host.	36
Configuring the Media Server for Windows Server 2003 (x86/x86_64), 2008 and 2008 R2 x64 OS media	36
Windows Media: Preparing Network Driver Directories	36
Windows Media: Hosting Windows Media on a Windows 2000 Server using a Share	37
Configuring the Media Server for Red Hat Linux or VMware ESXi OS media.	37
Configuring the Media Server for SUSE Linux or SUSE Enterprise Linux OS media.	38
Configuring the Media Server for Sun Solaris.	40
Steps to Create MRLs.	40
Media Resource Locator (MRL) Administration.	41
Editing MRLs.	41
Deleting MRLs.	42
Advanced Import Media Tool Information.	42
Configuring RAID on HP ProLiant Servers Before OS Provisioning	43
Supported Hardware	43
Supported Operating Systems	43
Capture a Baseline HP ProLiant RAID Configuration	45
Creating an HP ProLiant RAID Dynamic Server Group	46
Manually Specifying an HP ProLiant RAID Configuration	47
4 Defining Installation Profiles, Build Plans and OS Sequences.	49
OS Installation Profile Requirements.	49
Overview	49
Specifying Software for OS Provisioning.	51
Configuration Files	51
Sun Solaris Installation Profile Requirements	51
Red Hat Linux Installation Profile Requirements	52
VMware ESX Installation Profile Requirements	52
SUSE Linux Installation Profile Requirements	52
Microsoft Windows Installation Profile Requirements.	53
Sample Response File for Windows Server 2003.	53
Sample Response File for Windows Server 2008 x86	54
Sample Response File for Windows Server 2008 x64	56

Defining and Managing OS Installation Profiles	58
Defining an OS Installation Profile — Linux/Unix	59
Defining an OS Installation Profile — Windows	60
Hardware Signature Files for Windows	63
Modifying Existing OS Installation Profiles	65
Changing the OS Installation Profile Properties	65
Modifying How an OS Is Installed on a Server — Linux/Unix	66
Modifying How an Operating System Is Installed on a Server — Windows	66
Modifying the OS Installation Profile Packages	68
Viewing Change History for an OS Installation Profile	69
Deleting an OS Installation Profile	70
Creating Build Customization Scripts	71
Using Build Customization Scripts	71
Solaris Build Customization Scripts	72
The Sun Solaris Build Process	72
Requirements for Solaris Build Customization Scripts	74
Solaris Provisioning from a Boot Server on a Red Hat/SLES 10 Linux Server	74
Creating a Solaris Build Customization Script	75
Sample Solaris Build Customization Script	76
Linux Build Customization Scripts	76
Linux/Itanium Build Process	76
Requirements for Linux Build Customization Scripts	78
VMware ESX Build Process	79
VMware ESX Build Customization Scripts	79
Windows Build Customization Scripts	79
Windows Build Process (WinPE Boot Image)	79
Legacy Build Customization Script run.bat	80
Creating a Windows Build Customization Script (WinPE)	81
Sample run.cmd File	82
Defining Custom Attributes	83
Custom Attributes for Sun Solaris	84
Custom Attributes for Linux or VMware ESX	85
Using the boot_disk Custom Attribute to Specify the Boot Drive	86
Custom Attributes for Microsoft Windows	87
Adding Custom Attributes to OS Installation Profile (SA Web Client)	89
Adding Custom Attributes to OS Installation Profile (SA Client)	90
Creating OS Build Plans	90
Support for the OGFS Agent	91
Effect of the OGFS Agent on Server Lifecycle	92
What are OS Build Plans?	92
Baseline OS Build Plans	93
OS Build Plan Updates	94
Sample OS Build Plan Usage Instructions	94
OS Build Plan Requirements	94
Permissions	95
Custom Attribute Parameters and OS Build Plans	95
Copying a Baseline OS Build Plan	95

Viewing/Modifying OS Build Plans	96
Minimum Baseline OS Build Plan Modification	99
Creating OS Sequences	100
OS Sequence Contents	100
Defining an OS Sequence	101
5 Performing OS Provisioning	103
SA OS Provisioning-supplied CD Boot Images	103
Sun Solaris OS Provisioning Quick Start	103
Linux or VMware ESX OS Provisioning Quick Start	104
Microsoft Windows OS Provisioning Quick Start	105
Booting Servers Remotely	106
Booting from a CD	106
Network Booting a Linux or VMware ESXi Server	107
Intel-based Linux or VMware ESXi PXE Booting	107
Intel-based Linux or VMware ESXi OGFS Agent Booting	107
Booting a Red Hat Enterprise Linux Server in a Non-DHCP Environment	108
DHCP Custom Attribute	109
Booting a Red Hat Enterprise Linux Itanium 64-bit Server in a Non-DHCP Environment	110
DHCP Custom Attribute	112
Network Booting a Windows Server	112
Windows PXE Booting	113
Windows OGFS Agent Booting	114
Booting a Windows Server in a Non-DHCP Environment	115
Booting an Unmanaged Windows Server in a Non-DHCP Environment	115
DHCP Custom Attribute	116
Network Booting Sun Solaris SPARC-based Server	117
Loading OS Build Agents	118
Verifying That a Server is Ready for Operating System Installation	118
The Manage Boot Clients (MBC) Option	119
Requirements	119
Required Permissions	120
Installation	120
Using the Manage Boot Clients (MBC) Option	120
The MBC Web Interface	121
The MBC Form-Based Method (Web-based)	121
The MBC APX Command-Line Interface	121
Special Attributes for the CLI and CSV Input Form	122
CSV Input Files	123
Special Attributes for DHCP Reconfiguration	124
iLO Integration	124
Installing (Provisioning) an Operating System	125
The Unprovisioned Servers List	125
Using an OS Build Plan for OS Provisioning	127
Using an OS Sequence for OS Provisioning	129
HP-UX Provisioning	130

Prerequisites	130
DHCP Setup	131
Ignite Setup on the SA Core	131
APXs	131
Customer Configuration Subfolders	132
Permissions	132
Installing an Operating System on HP-UX Servers	133
Creating a Custom Configuration	133
Boot Target	138
Provision the Target Servers	138
Deleting Custom Configurations	144
Glossary	145
Useful Links	146
Troubleshooting	146
Model Base Packages Functionality	150
Model Base Packages Script Usage	150
Reprovisioning a Managed Server	151
Recovering When an OS Build Agent Fails to Load During PXE Boot	152
Index	153

1 OS Provisioning Concepts

SA OS Provisioning provides the ability to provision operating system baselines onto bare metal and virtual servers quickly, consistently, and with minimal manual intervention. Bare metal and virtual server OS provisioning is a key part of the overall process of getting a server into production.

The benefits of SA OS Provisioning include:

- **Integration with other SA features**

Because SA OS Provisioning is integrated with the suite of SA automation capabilities, including patch management, software management, and distributed script execution, hand-offs between IT groups are seamless. SA ensures that all IT groups are working with a shared understanding of the current state of the environment, which is an essential element of delivering high-quality operations and reliable change management.

- **Update server baselines without re-imaging**

Unlike many other OS provisioning solutions, systems provisioned by SA can be easily changed after provisioning to adapt to new requirements. The key to this benefit is SA's use of templates and an installation-based approach to provisioning.

- **Flexible architecture designed to work in many environments**

SA OS Provisioning supports many types of servers, networks, security architectures, and operational processes. SA works well in CD (Linux and Windows provisioning) or network-boot environments (both DHCP and non-DHCP), with scheduled or on-demand workflows, and across a large variety of hardware models. This flexibility ensures that you can provision operating systems to suit your organization's needs.

SA automates the entire process of provisioning a comprehensive server baseline, which typically consists of the following tasks:

- Creating OS Build Plans or OS sequences that define a server build policy, including Software Policies, Patch Policies, Device Groups, and Remediation Policies.
- Installing a base operating system and default OS configuration using an OS Build Plan or OS Sequence.
- Applying the latest set of OS patches, the exact list depends on the applications running on the server.
- Installing widely-shared system software such as Java Virtual Machines.
- Executing pre-installation or post-installation scripts that configure the system with values such as a root password.

SA OS Provisioning integrates with your operating system vendors' native installation technologies, specifically:

- WinPE and Windows setup answer files: `unattend.txt`, `unattend.xml`, `sysprep.inf`
- Red Hat/VMware Kickstart
- SuSE YaST (Yet another Setup Tool)

- Solaris Jumpstart
- Static network (non-DHCP) provisioning (for Red Hat Enterprise Linux and Windows) using SA-supplied CD/DVD ISO boot images.
- Enabling separation of duties between data center staff and systems administrators.
- Providing a model-based approach — in which you create a *standard build* in SA which can then be installed on many systems.

You can provision an operating system on:

- Servers in SA's *Unprovisioned Server* pool that have no operating system installed.
- SA Managed Servers with a Linux or Solaris SPARC OS platform as well as Linux and Solaris SPARC (reprovisioning).

The OS Provisioning Process

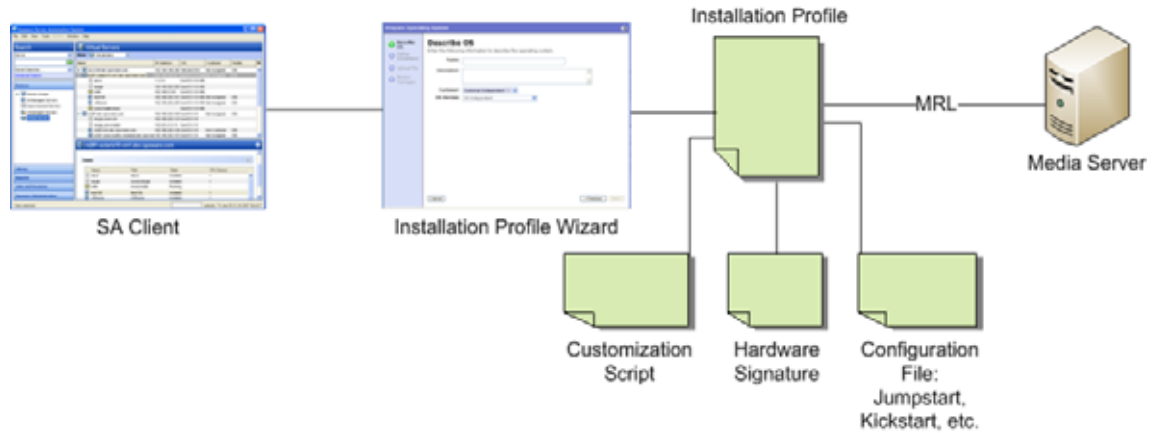
The SA OS Provisioning process consists of certain preparatory tasks including:

- Installing and configuring the required OS Provisioning components including the following:
 - Media server repository preparation including creating Media Resource Locators (MRLs) for use with *OS Sequences* (not required for *OS Build Plans* where media location is specified in the plan itself).
 - Uploading licensed OS media to the Media Server
 - DHCP server management using the `/opt/opsware/dhcpd/sbin/dhcpdtool`
- If you are using *OS Sequences*, creating *OS Installation Profiles* for the operating systems to be provisioned. The Installation Profiles specify which operating system is to be installed and how it is to be configured and where the operating system media is located on the Media Server (using an MRL).

Associated with the Installation Profile are:

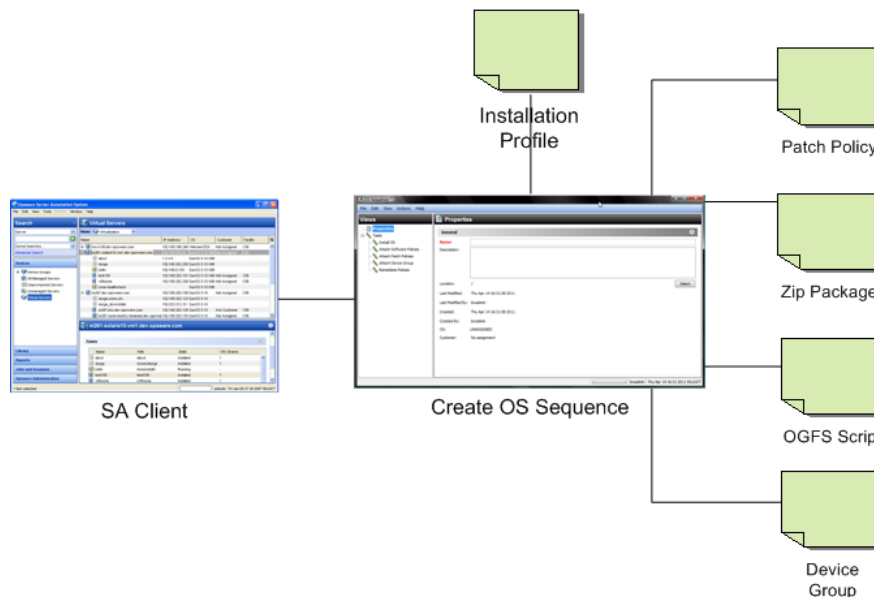
- Operating system-specific installation configuration files such as Kickstart (Linux), Jumpstart (Solaris), and `unattend.txt` or `.xml` (Windows).
- Build Customization Scripts that allow you to manage each operating system installation from the network connection to SA Agent installation.
- Custom Attributes that allow you to define server-specific custom attributes that set specified parameters and named data values. You can write scripts that use these parameters and data values to control network and server configuration, notifications, and CRON script configuration.

Figure 1 Create OS Installation Profile



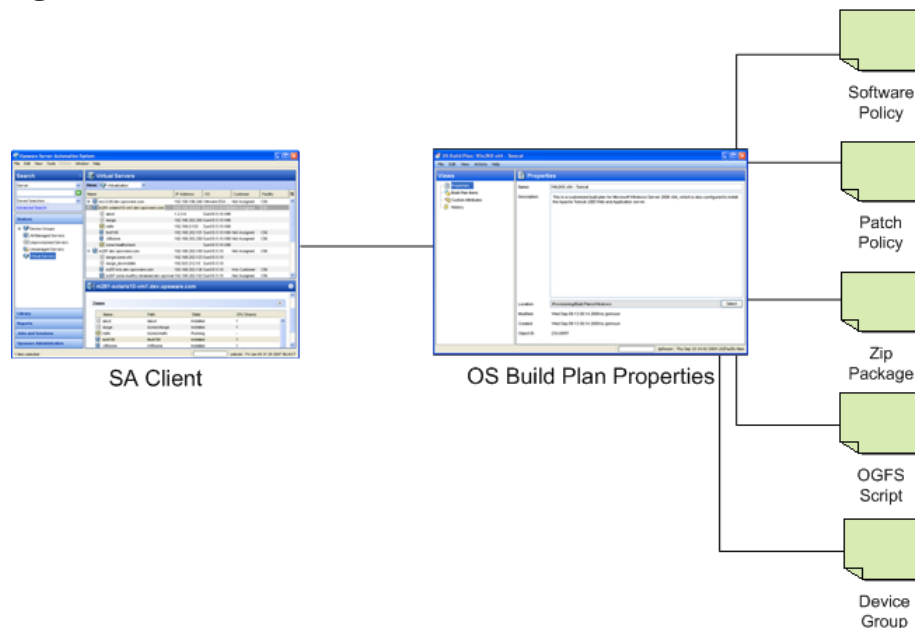
- Creating *OS Sequences* that specify the OS Installation Profile to use, device groups, and remediation policies. You must use OS Sequences in conjunction with OS Installation Profiles.

Figure 2 Create OS Sequence



- Creating *OS Build Plans*. If you are using OS Build Plans, no Installation Profile is required. OS Build Plans combine and enhance the functionality of OS Installation Profiles and OS Sequences,

Figure 3 Create OS Build Plan



After you have created your OS Provisioning configuration files, the process for provisioning new servers typically includes tasks similar to the following:

1 Preparation

- Physically prepare the server for operation and connect it to a network that can communicate with SA.
- In some cases, you must prepare the server hardware for OS Provisioning.
See [Hardware Preparation](#) on page 25.
- (*OS Sequences only*) OS Installation Profile(s) defined and available.
- OS Build Plans (Windows and Linux only) and or OS Sequences defined and available.

2 Booting the server to be provisioned

Power on and boot the server to be provisioned using one of the following boot methods:

- Use a bootable CD, or DVD provided by SA.



The bootable CD or DVD is not required for Intel-based servers that support PXE/WinPE/WinPE-OGFS or Unix servers as these servers can be remotely booted over a network.

- For servers that can be booted over the network, powering on the server causes the server to initiate its network boot process.

For more information about booting servers remotely, see [Intel-based Linux or VMware ESXi PXE Booting](#) on page 107 and [Intel-based Linux or VMware ESXi OGFS Agent Booting](#) on page 107, [Windows PXE Booting](#) on page 113 or [Windows OGFS Agent Booting](#) on page 114 or [Network Booting Sun Solaris SPARC-based Server](#) on page 117, and, for HP ILO servers, [The Manage Boot Clients \(MBC\) Option](#) on page 119.

- 3 After the server boots successfully and the SA Build Agent or SA OGFS Agent has been installed, the server appears in the SA Client in the Unprovisioned Server list and is ready for operating system installation.

See [Verifying That a Server is Ready for Operating System Installation](#) on page 118.

- 4 **Install the Operating System (Provision)**

- Select a server that has been booted with an *OGFS Agent* and select an *OS Build Plan* or
- Select a server that has been booted with an *OS Build Agent* and select an appropriate *OS Sequence* for the operating system and configuration you want to install.

- 5 Start the OS Provisioning job.

SA OS Provisioning Components

SA OS Provisioning is an optional feature that can must be installed for each SA Core where operating system provisioning is to be performed. For information about installing the required OS Provisioning components, see the *SA Simple/Advanced Installation Guide*.

SA OS Provisioning uses the following components and features.

- The OS Build Agent
- OGFS Agent (a specialized SA Server Agent)
- The Build Manager
- The Media Server
- The Boot Server

The OS Build Agent

Used with *OS Sequences*. Similar to the SA Server Agent, the OS Build Agent is a simplified agent whose function is to run commands as instructed by the Build Manager. Newly registered servers with installed OS Build Agents appear in the SA Client Unprovisioned Server list.

Booting a new server for the first time loads an OS Build Agent on the server; however, the server does not have the target operating system installed and might not have access to disk resources. SA can still communicate with the server and perform commands on it remotely because the OS Build Agent is running a limited operating system that is loaded into memory.

The OS Build Agent performs the following functions:

- Registers the server with SA when the OS Build Agent starts.
- Listens for command requests from SA and performs them.
- Performs commands even though a target operating system is not installed.

OGFS Agent

Used with *OS Build Plans*. The OGFS Agent, a specialized SA Server Agent, is part of certain SA-supplied operating system boot images. When an unprovisioned server is PXE booted, the OGFS is installed, the server is registered with the SA Core, an agent certificate is obtained from the core, and the server appears in the SA Client Unprovisioned Server list. The OGFS Agent uses Core's OGFS functionality to complete Agent tasks.

The Build Manager

OS Sequences only. The build manager performs several functions:

- Manages newly registered OS Build Agents.
- Coordinates scripts that gather hardware inventory from OS Build Agents.
- Coordinates the scripts that perform the operating system installation with the OS Build Agent.
- Communicates with the OS Build Agents using a simple protocol.

The Media Server

The Media Server is installed as part of a typical SA Core installation when you specify that you want to install the OS Provisioning components. In order to provision operating systems, you must first upload a valid copy of the operating system's installation media to the Media Server. During OS Provisioning, SA will use the copy of the operating system installation media on the Media Server to do the provisioning.

SA provides file servers that can share operating system media using NFS and Samba if you do not have existing NFS/Samba servers that you want to use or are not familiar with configuring these servers.

The Boot Server

The Boot Server listens for broadcast requests from new servers in the server pool and responds using DHCP. Network booting requires DHCP/BOOTP, TFTP, and PXE (x86).

Build Customization Scripts

OS Sequence only. OS Provisioning build customization scripts provide hooks into the build process that allow you to modify operating system installations at specific points. These hooks call a single build customization script at the appropriate time in the operating system installation process.

Because each build customization script is specific to the operating system it installs, build customization and installation vary by operating system. Before you can use a build customization script as part of an operating system installation profile, you need to create the build customization script and import it into the SA Client.

How the OS Build Agent Locates the Build Manager

OS Sequence only. How the OS Build Agent locates the Build Manager depends on the boot method.

WinPE

- SA retrieves DHCP options containing the agent gateway IP address and Build Manager port, or
- The Build Manager is located by loading the configuration file:
`/opt/opsware/boot/tftpboot/DHCPOptions.ini`
which contains the OS Provisioning settings specified during SA installation.
- If the processes above fail, SA defaults to the hostname `buildmgr` on port `8017`.

Linux:

Linux x86 locates the Build Manager using kernel arguments supplied at PXE boot time. These are configured during the SA installation and stored in the file

```
/opt/opsware/boot/tftpboot/pxelinux.cfg/default
```

Linux IA64:

Linux IA64 locates the Build Manager using kernel arguments supplied at PXE boot time. These are configured during the SA installer and stored in the file

```
/opt/opsware/boot/tftpboot/elilo.conf
```

Solaris

For Solaris OS Provisioning, the JumpStart build script runs the OS Build Agent, which contacts the Build Manager (via the Agent Gateway in the core). The Solaris `begin` script attempts to locate the Build Manager in the following ways:

- By using information that the SA DHCP server provided
- By looking for the host name `buildmgr` in DNS as configured by the DHCP server

You can override the way that the OS Build Agent contacts the Build Manager by specifying a boot argument at the prompt when you boot a new Solaris server, for example:

```
ok boot net:dhcp - install buildmgr=buildmgr.example.com:8017
```

```
ok boot net:dhcp - install buildmgr=192.168.1.15:8017
```

Non-DHCP Environments

In both Windows and Red Hat non-DHCP environments, SA locates the Build Manager using the network configuration specifications you provide. See [Booting a Red Hat Enterprise Linux Server in a Non-DHCP Environment](#) on page 108 and [Booting a Windows Server in a Non-DHCP Environment](#) on page 115.

2 OS Provisioning Requirements

To prepare for OS Provisioning, authorized staff should determine and record the standard configuration of each operating system to be provisioned as well as the required utilities, drivers, and agents.

SA Core Setup Requirements

There are a number of configuration tasks that are performed before or after installation of the SA Core. For more information, see the *SA Simple/Advanced Installation Guide*. In addition, ensure that the following requirements are met.

Permissions

Your SA/System or OS Provisioning Administrator must have granted you a specific set of feature permissions in order for you to set up OS Provisioning and/or perform provisioning. You must also have the permissions to access the servers associated with SA customers, facilities, or server groups.

For more information, see the Permissions Reference appendix in the *SA Administration Guide*.

Network Setup for OS Provisioning

- It is essential that you correctly configure any network switch ports used for OS Provisioning. These switch ports must have PortFast mode enabled and must be set for speed/duplex auto-negotiation. While provisioning using manually configured interface speed and duplex settings is possible for Solaris-based and Red Hat Linux-based boot images, It is recommended that you use auto-negotiation as it has been found to work the most consistently. See “Open Ports” in the *SA Simple/Advanced Installation Guide*.
- You should configure the OS Build Agent to connect to the OS Provisioning Build Manager using the IP address as opposed to the DNS name.
- If you are using an SA Boot/DHCP server for IP address management, ensure that the server is configured appropriately using the `/opt/opsware/dhcpd/sbin/dhcpdtool` script.
- If you must use DNS names, you must specify the DNS name during the SA Installer interview and save it in the Response File (see the `boot_server.buildmgr_host` parameter). You must also configure DNS so that the servers being provisioned can resolve the Build Manager host. The hostnames of all OS Provisioning Media Servers must also be resolvable.

- If you will be performing a Solaris PXE boot of an Intel-based server with an SA Boot Server that is hosted on a server running Linux, ensure that NFSv2 is enabled and that NFSv3 or NFSv4 is disabled.". For more information, see [Custom Attributes for Sun Solaris](#) on page 84.
- If you will be performing a PXE boot of an Intel-based server with an SA Boot Server that is hosted on a server running Solaris and have NFSv4 enabled, you may experience unexpected issues, for example inability to access a target server over OGFS. If so, you should disable NFSv4 on the Solaris server.

Firewall Considerations

The following operating systems come with default firewall settings that must be modified during the operating system installation process in order to allow the SA Server Agent to be properly installed and configured on the target server.

- VMware ESX Server 4.1
- Windows Server 2008, Windows Server 2003 x64 and Window Server 2003 R2
- Windows XP SP2

OS Provisioning makes minor modifications to the firewall configurations on the managed server such that communication between the SA core and the Server Agent is not blocked.

Windows Server 2008, Windows Server 2003 SP1 and Windows XP SP2 Firewall Settings

For Windows Server 2008, Windows 2003 SP1 and Windows XP SP2, in order for OS Provisioning and ongoing management to succeed, SA must ensure that the Windows firewall settings are configured to bypass the default "Security Out Of the Box" experience and allow communication over the SA ports. Thus the OS Provisioning process updates the Windows Firewall settings in the `unattend.txt`, `unattend.xml`, or `sysprep.inf` answer file as necessary for provisioning and management to work.

OS Provisioning looks for the following Windows Firewall configurations in `unattend.txt`, `unattend.xml`, or `sysprep.inf`:

- There is no Windows firewall configuration.
- There is a Windows firewall configuration, but it does not allow the ports needed by SA.
- There is a Windows firewall configuration that does allow the ports needed by SA (no changes will be made).

In any of the cases, after running an OS Sequence and installing the operating system (and agent), any predefined firewall settings remain in tact, with the exception that the SA Server Agent will have been installed and all of its required ports will have been opened.

Red Hat Enterprise Linux 4, 5 and 6 Firewall Settings

For Red Hat Enterprise Linux 4, 5 and 6, the following line in your `ks.cfg` profile will enable the firewall and allow the Server Agent to function correctly:

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

Suse Linux Enterprise Server Firewall Settings

For Suse Linux Enterprise Server 10, the following lines in your `autoyast.xml` profile will enable the firewall and allow the SA Agents to function correctly.

```
<firewall>
  <FW_ALLOW_FW_BROADCAST_DMZ>no</FW_ALLOW_FW_BROADCAST_DMZ>
  <FW_ALLOW_FW_BROADCAST_EXT>no</FW_ALLOW_FW_BROADCAST_EXT>
  <FW_ALLOW_FW_BROADCAST_INT>no</FW_ALLOW_FW_BROADCAST_INT>
  <FW_DEV_DMZ></FW_DEV_DMZ>
  <FW_DEV_INT></FW_DEV_INT>
  <FW_FORWARD_ALWAYS_INOUT_DEV></FW_FORWARD_ALWAYS_INOUT_DEV>
  <FW_FORWARD_MASQ></FW_FORWARD_MASQ>
  <FW_IGNORE_FW_BROADCAST_DMZ>no</FW_IGNORE_FW_BROADCAST_DMZ>
  <FW_IGNORE_FW_BROADCAST_EXT>yes</FW_IGNORE_FW_BROADCAST_EXT>
  <FW_IGNORE_FW_BROADCAST_INT>no</FW_IGNORE_FW_BROADCAST_INT>
  <FW_IPSEC_TRUST>no</FW_IPSEC_TRUST>
  <FW_LOG_ACCEPT_ALL>no</FW_LOG_ACCEPT_ALL>
  <FW_LOG_ACCEPT_CRIT>yes</FW_LOG_ACCEPT_CRIT>
  <FW_LOG_DROP_ALL>no</FW_LOG_DROP_ALL>
  <FW_LOG_DROP_CRIT>yes</FW_LOG_DROP_CRIT>
  <FW_MASQUERADE>no</FW_MASQUERADE>
  <FW_PROTECT_FROM_INT>no</FW_PROTECT_FROM_INT>
  <FW_ROUTE>no</FW_ROUTE>
  <FW_SERVICES_DMZ_IP></FW_SERVICES_DMZ_IP>
  <FW_SERVICES_DMZ_RPC></FW_SERVICES_DMZ_RPC>
  <FW_SERVICES_DMZ_TCP></FW_SERVICES_DMZ_TCP>
  <FW_SERVICES_DMZ_UDP></FW_SERVICES_DMZ_UDP>
  <FW_SERVICES_EXT_IP></FW_SERVICES_EXT_IP>
  <FW_SERVICES_EXT_RPC></FW_SERVICES_EXT_RPC>
  <FW_SERVICES_EXT_TCP>1002</FW_SERVICES_EXT_TCP>
  <FW_SERVICES_EXT_UDP>1002</FW_SERVICES_EXT_UDP>
  <FW_SERVICES_INT_IP></FW_SERVICES_INT_IP>
  <FW_SERVICES_INT_RPC></FW_SERVICES_INT_RPC>
  <FW_SERVICES_INT_TCP></FW_SERVICES_INT_TCP>
  <FW_SERVICES_INT_UDP></FW_SERVICES_INT_UDP>
  <enable_firewall config:type="boolean">true</enable_firewall>
  <start_firewall config:type="boolean">true</start_firewall>
</firewall>
```

Requirements for Provisioned Servers

Supported Operating Systems and Media for OS Provisioning



For a complete listing of all platforms supported for OS Provisioning, see the *SA Support Matrix* that is provided in the documentation directory of the distribution media or downloaded from the HP Self Solve portal. The document on the HP Self Solve portal is updated regularly and contains the most recent information about supported platforms.

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

Supported Boot Media

SA OS Provisioning works with:

- A CD-ROM for Windows via the WinPE preinstallation environment
- A CD-ROM for Linux
- Network booting for all supported operating systems.



Non-network booting is not supported for Sun Solaris (SPARC and x86).

SA OS Provisioning-supplied CD Boot Images

SA OS Provisioning provides several service operating system boot CD images (ISOs) that you can use to record to CD. These ISO images can also be configured in virtual machine CD-ROM drives or mounted using iLO Virtual Media or similar technology. Use the SA Client Export utility to download the required image(s) and burn boot CDs.

WinPE

These files are located in the SA Software Library Folder and are named using the format:

```
/Opsware/Tools/OS Provisioning/WinPE/OPSWwinpe<arch>.iso
```

Images currently available are:

- OPSWwinpex86.iso
- OPSWwinpex86-ogfs-enabled.iso
- OPSWwinpex64.iso
- OPSWwinpex64-ogfs-enabled.iso

Linux

These files are located in the SA Software Library Folder. Images currently available are:

- /Opsware/Tools/OS Provisioning/HPSA_linux_boot_cd.iso
- /Opsware/Tools/OS Provisioning/HPSA_linux_boot_cd_IA64.iso

Itanium-Based Systems

As of SA 9.10 and later, OS Provisioning supports provisioning only on Red Hat Enterprise Linux Itanium systems.

Suse Linux Enterprise Server and Windows Itanium-based systems are *not* supported.

Solaris Servers



In order to perform PXE booting of a VMware ESX Solaris 64-bit VM, the minimum required RAM is one (1) GB.

OS Provisioning includes a DHCP-based JumpStart configuration that makes the complexity of JumpStart transparent to the end user.

For example, unlike typical JumpStart systems, OS Provisioning does not require configuration updates to the JumpStart server for each installation that you provision. Instead, SA provides an OS Installation Profile for supported version of the Solaris operating system.

Solaris OS Provisioning generally follows the typical provisioning process.



If you will be performing a PXE boot of an Intel-based server with an SA Boot Server that is hosted on a server running Solaris and have NFSv4 enabled, you may experience unexpted issues, for example inability to access a target server over OGFS. If so, you should disable NFSv4 on the Solaris server.

See [Solaris Build Customization Scripts](#) on page 72 for more information about Solaris build customization scripts.

SPARC SUN4U Servers

In order to provision a bare metal SPARC SUN4U server with any of the supported SPARC Solaris versions, the server must support *Solaris 10 U9*.

Windows Servers

Windows system administrators can perform unattended, scripted installations as well as WinPE-based image installations of Windows Server 2008, Windows Server 2008 R2 x64, Windows Server 2003, Windows XP Professional, and Windows 2000.

This installation-based approach allows system administrators to adapt to variations in hardware. OS Provisioning can use the information about correct hardware-specific software and drivers contained in a server's hardware signature file.

WinPE Memory Requirements

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

Virtualization Support — VMware ESX, ESXi and Solaris 10



SA supports provisioning VMware ESXi 4.1 using OS Build Plans only.

SA supports VMware ESXi 4.1 as an operating system that can host virtual servers.

- Using an OS Sequence, you can provision VMware ESX on a bare metal server as a VMware ESX hypervisor.
- Using an OS Build Plan, you can provision VMware ESXi 4.1 on a bare metal server as a VMware ESXi 4.1 hypervisor
- You can provision any VMware ESX/ESXi-supported guest operating system on a VMware ESX/ESXi virtual server host just as you would provision a physical server.

Because the OS Provisioning process for VMware ESX (using an OS Sequence) / ESXi 4.1 (using an OS Build Plan only) follows the same general process as provisioning Linux (with some minor differences), you provision VMware ESX/ESXi 4.1 as you would generic Linux. Any differences in the provisioning process between Linux and VMware ESX/ESXi 4.1 are documented separately.

HP-UX Servers

The HP-UX Provisioning feature in SA enables you to install HP-UX on bare metal servers using golden image archives. You can model your HP-UX installation with customized configurations to provision multiple servers using HP's native unattended provisioning technology, Ignite-UX. HP-UX Provisioning in SA does not require separate dedicated Ignite-UX setup, instead SA core running with RedHat Linux can be used with Ignite-UX setup. HP-UX Provisioning feature is also supported on multi-master mesh and satellite environment.

Using HP-UX Provisioning feature in SA, you can:

- Perform network-based unattended installations using Ignite golden images
- Define Ignite attributes values different than specified in golden image for customized provisioning
- Bundle additional SA software policies to remediate as part of HP-UX installation
- Segment custom configurations in multi-customer environment on multi-mesh environment with permissions escalations.

Table 1 Supported HP-UX Platforms and Systems

Supported	Not Supported
Bare metal HP-UX provisioning on Integrity servers only	HP 9000 (PA-RISC)
All Integrity standalone, blade, nPar/vPar/HPVM	Support for provisioning other operating systems such as Windows and Linux
Provisioning HP-UX 11.23 and 11.31 only	Provisioning HP-UX 11.11 or earlier
Provisioning using golden images	Alternate methods to provision HP-UX servers such as software depot-based install
Network boot support only	
Multi-customer environment	
HP-UX configuration selection based on customer permissions granted	
Configuration customization:	
Configurations isolated by customer	
Ability to upload Ignite image files as part of operating system profiles	
Bundling additional SA software policies to remediate	
SA Core platforms - RHEL (AS4-x64), Solaris, SUSE	

OS Build Plan Requirements

- As of SA 9.10. OS Build Plans are available for the Windows Server (2003, 2008 and 2008 R2 x64), Red Hat Enterprise Linux (Server 5 and 6) and VMware ESXi 4.1 platforms.

- Provisioning VMware ESXi 4.1 is supported only using OS Build Plans. Other ESXi versions *are not supported* whether using OS Build Plan or OS Sequence.
- OS Build Plans require Automation Platform Extensions (APX) to complete certain tasks. In order for these APXs to operate, you must have the latest Adobe Flash Player installed on all machines from which you run the SA Client and OS Build Plans.
- SA provides a set of baseline OS Build Plans that you copy and use as a template for your own Build Plans. These Build Plans are installed by default during SA installation or upgrade. For out-of-band OSBP updates, you will be required to download the new OSBPs from the HP Live Network (HPLN). The baseline Build Plans are provided with a `readme.html` file that describes their installation and usage. This file is also available in SA Online Help. These default baseline OS Build Plans have been tested and are known to work. They simply need to be copied and adapted to your environment (location of the Media Server, required scripts, required reboots, etc.). For more information, see [Creating OS Build Plans](#) on page 90.

Hardware Preparation

Before you use OS Provisioning to install an operating system, the target server must meet certain requirements which can vary according to the operating system being provisioned.

Windows Hardware Preparation Requirements

Before you provision the Windows operating system, you must prepare the hardware by performing the following tasks:

- OS Provisioning supports RAID configuration during provisioning, however you must complete certain configuration steps. For more information, see [Configuring RAID on HP ProLiant Servers Before OS Provisioning](#) on page 43.
- If there is a RAID controller installed, you may have to extend the Windows operating system media distribution (provide third party RAID drivers) based on hardware vendor-specific requirements. The Microsoft Windows operating system media (depending on the version of Windows) may not include the necessary drivers for many RAID controllers. Certain newer types of SATA controllers may also require additional drivers.
- If you use a WinPE-based PXE or CD-ROM boot image to install the Windows operating system, disk partitioning is performed as part of the operating system installation. You can control the disk partitioning by editing the OS Installation Profile. (For more information about creating installation profiles, see [Defining and Managing OS Installation Profiles](#) on page 58.) Partitioning can also be controlled as an OS Build Plan task. See [Creating OS Build Plans](#) on page 90.
- If you use a WinPE-based PXE or CD-ROM boot image and you are using a RAID or SATA controller, you may need to supply an operating system-specific Build Customization Script for OS Sequence based provisioning. These scripts enable you to load necessary hardware drivers before the operating system installation commences. For more information about Build Customization Scripts, see [Creating Build Customization Scripts](#) on page 71. For OS Build Plan-based provisioning, you may be able to load necessary hardware drivers using OS Build Plan tasks. See [Creating OS Build Plans](#) on page 90.

Sun Solaris Hardware Preparation Requirements

To provision Solaris to a server, the hardware must meet the following requirements:

- The server must have a DHCP-capable PROM (older servers can be upgraded to DHCP-capable PROM).
- The server must be part of the SUN4U system architecture (platform group).

VMware ESXi Hardware Preparation Requirements

- There are no special hardware requirements for VMware ESXi, however, if you have RAID drives installed, you must prepare the hardware by configuring valid, logical drives for RAID.
- Refer to the VMware documentation for additional VMware ESXi hardware requirements.

Red Hat Linux Hardware Preparation Requirements

You must change the configuration of the managed switch for Red Hat Linux to enable PortFast. If this isn't done, when the Red Hat Linux installer attempts to use NFS to mount the media, the DHCP request could time out. (This problem is fixed in the packages listed in the Red Hat advisory RHEA-2004:518-06.)

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 Hardware Support to a Linux or VMware ESX Build Image](#) on page 28 for more information.

NIC Support in PXE Images for Linux

OS Provisioning supports booting 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 provides both CD boot and PXE boot images. As of SA 9.10, CD boot images include a boot image that contains the `bootnet.img` CD for Red Hat Enterprise Linux 5.6, and 6.0. The image uses the `syslinux.cfg` and `boot.msg` files, however, the kernel and `initrd.img` are identical to the files on the Linux OS media.

For PXE images, SA 9.10 and later provides the following images:

- **linux5**

This PXE image is based on Red Hat Enterprise Linux 5.6. It provides the same hardware compatibility as Red Hat Enterprise Linux 5.6. Before applying these images, you should confirm your servers' hardware compatibility with Red Hat Enterprise Linux 5.6 by referring to the Red Hat Compatibility Guide which can be found at:

<https://hardware.redhat.com/hwcert/list.cgi>

- **linux6 / linux6-ogfs**

The `linux6` and `linux6-ogfs` options on the SA 9.10 boot menu use boot images that are based on Red Hat Enterprise Linux 6 to bring servers into the SA server pool. Before applying these images, you should confirm your servers' hardware compatibility with Red Hat Enterprise Linux 6 by referring to the Red Hat Compatibility Guide which can be found at:

<https://hardware.redhat.com/hwcert/list.cgi>

NIC Support in Solaris Boot Images

SA provides various versions of Solaris x86 boot images (U4, U5, U9). For Solaris SPARC, U9 is supported.

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

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

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 2 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)
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

Table 2 NIC Drivers Included with the Windows Boot Image (cont'd) (cont'd)

Driver Name	Description
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)

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 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 operating system, 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 operating system.

- **A Linux Build Image that installs the target operating system:** SA uses this type of Linux Build Image to install the target Linux operating system on servers.

To add new hardware support to a Linux Build Image, you must recompile the kernel and modules, and insert the modules into the `initrd.img` file and replace the kernel if it changed.

The Linux Boot Images are located on the OS Boot Server host in the following directories:

Red Hat Linux 3 and 5

```
/opt/opsware/boot/kickstart/rhel*/RedHat/base
```

Red Hat Linux 5

```
/opt/opsware/boot/kickstart/rhel*/images/
```

There are two versions of each boot image. For example, the `rhel30` directory contains the Red Hat Enterprise Linux AS 3 boot image for x86 32- and 64-bit architectures, and the `rhel3ia` directory contains the boot images for Itanium architecture.

The Linux Build Images are located inside the OS installation media. The media is found in the media server.

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.

3 Operating System Provisioning Setup

To prepare for OS Provisioning, authorized staff should determine and record the standard configuration of each operating system to be provisioned as well as the required utilities, drivers, and agents. System administrators can then use OS Provisioning to install the operating systems, configure networking, and install other software.

Before using SA OS Provisioning you must complete a number of preparatory tasks including:

- Confirming that required permissions are set for OS Build Plan and/or OS Sequence *management* and/or OS Build Plan/OS Sequence *execution*.
- Confirming that the network is configured as required for OS Provisioning.
- Preparing hardware to be provisioned as required.
- Configuring the SA Media Server for the operating systems you will provision.
- Uploading licensed operating system media to the SA Media Server.
- (*OS Sequence-based provisioning only*) Creating Media Resource Locators (MRLs) that identify the location of the media during provisioning.
- Configuring optional HP RAID configuration capture.
- (*OS Sequence-based provisioning only*) Creating optional Build Customization scripts for the operating systems you will provision.
- Creating optional Custom Attributes for the operating systems you will provision.
- Creating OS Build Plans or OS Sequences that specify the order of provisioning tasks and can include optional configuration information, Software Policies, Windows Patch policies, Static Device Groups. For OS Build Plans, additional objects such as Server Scripts, OGFS Scripts and ZIP packages are also supported.
- (*OS Sequence-based provisioning only*) Creating Installation Profiles for the operating systems you will provision. You can also optionally add custom attributes and build customization scripts to the Installation Profiles.

The following section summarizes those tasks and provides pointers to detailed instructions for completing the tasks.

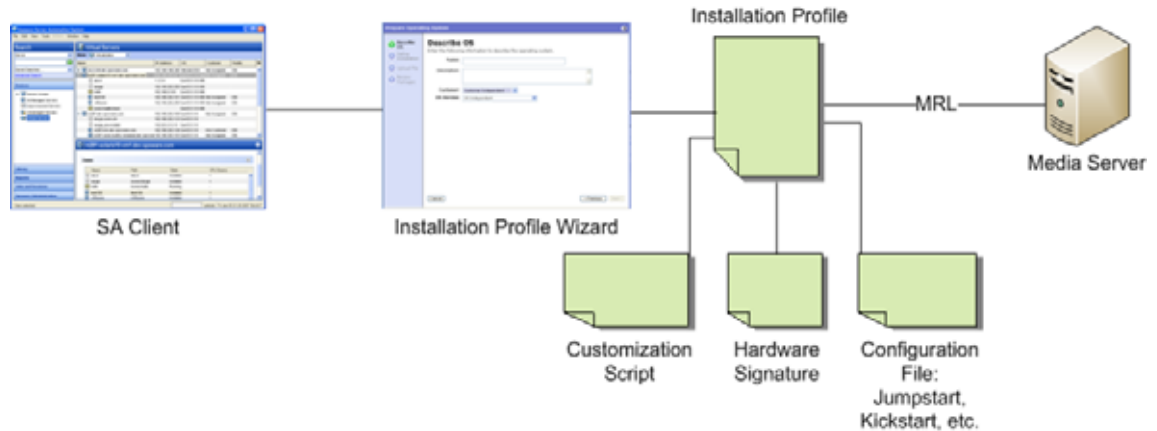
OS Provisioning Setup Task Summary

The required OS Provisioning setup tasks, typically performed by an OS Provisioning or SA Administrator, include:

- 1 Installing the OS Provisioning Components. This task should have been completed when SA was installed. If not, contact your SA or System Administrator. See the *SA Simple/Advanced Installation Guide*.
- 2 Configure the SA Boot/DHCP server using the `/opt/opsware/dhcpd/sbin/dhcpdtool` script. This script is installed with the SA Boot server component.

- 3 Installing the Media Server. This task should have been completed when SA was installed. If not, contact your SA or System Administrator. See the *SA Simple/Advanced Installation Guide*.
- 4 Ensuring that you have the correct permissions for OS Build Plan/OS Sequence *management* and/or OS Build Plan/OS Sequence *execution*. To obtain these permissions, contact your SA/System Administrator. For more information about which permissions are required, see the Permissions Reference appendix in the *SA Administration Guide*.
- 5 Setting up the Media Server:
 - a Copying your licensed operating system media or images to the Media Server NFS/Samba share. After the media or images are copied to these shares, ensure that at minimum Read permission is given to these shares. For OS Build Plan-based provisioning, the Windows media can be either be stored in a Samba share or in a Windows Domain share.
 - b (*OS Sequence-based provisioning only*) Using the `import_media` script tool to create the Media Resource Locators (MRLs) for the operating system media. The `import_media` script is installed with SA's Software Repository component. This step is required to create OS Installation Profiles and OS Sequences.
- 6 [*Optional*] Setting up HP RAID capture which enables provisioning RAID configured servers.
- 7 (*OS Sequence-based provisioning only*) Creating Build Customization scripts that allow you to modify operating system installations at specific points.
- 8 [*Optional*] Creating Custom Attributes that allow you to set certain parameters and named data values.
 - For *OS Build Plan-based provisioning*, since the OS Build plan is not attached to a server, these custom attributes should be added to either the target server itself or to any objects that inherit from the server, such as facility, customer, device group, etc.
 - For *OS Sequence-based provisioning*, custom attributes can also be added to OS Installation Profiles.
- 9 Creating OS Build Plans or OS Sequences which allow you to specify provisioning tasks, the Installation Profile to use (*OS Sequences only*), configuration file/script to use (*OS Build Plans only*), reboots, etc. While this task can be done by the OS Provisioning Administrator, it can also be delegated to users who perform provisioning.
- 10 (*OS Sequence-based provisioning only*) Creating OS Installation Profiles in which you specify:
 - The location of resources (MRLs), configuration files (Jumpstart, KickStart, YAST2, and Windows unattended install files)
 - The OS Sequence
 - How provisioning is to be performed (Build Customization scripts and Custom Attributes)
 - Software packages to be provisioned with the operating system
 - Hardware signature files

Figure 4 Creating an OS Provisioning Installation Profile



The following sections contain instructions to set up SA OS Provisioning for supported platforms.

Setting Up the Media Server

The Media Server is the repository for the operating system media (images) that SA uses during OS Provisioning. You must prepare the Media Server by uploading your images to the Media Server.

For OS Build Plan-based provisioning, after the media is uploaded to the Media Server shares, the Media Server location and OS installation directories are typically specified in a script's step parameter field. See [Creating OS Build Plans](#) on page 90.

For OS Sequence-based provisioning, after the media is uploaded to the Media Server shares, you must create Media Resource Locators (MRLs) by running a script tool called `import_media`. This tells SA where to find the images on the Media server for OS Sequence-based provisioning. For more information about the `import_media` script, see [Creating Media Resource Locators \(MRLs\)](#) on page 33. During provisioning, the MRL is used to locate a specified image and install the new operating system on an unprovisioned server.

The Media Server provides access to images over a network using NFS for Linux, VMware ESXi, and Solaris systems and SMB/CIFS for Windows systems. For OS Build Plan-based provisioning, Windows media can also be stored in Windows or Windows domain share.

A single copy of the operating system media on the Media Server can be used to provision multiple servers as long as you have valid licenses and/or license keys.

Creating Media Resource Locators (MRLs)

You must perform several steps to create the Media Resource Locators (MRLs). The `import_media` tool is first used to import your operating system media.

Import Media Tool Prerequisites

- Before you run the Import Media tool, the operating system media that you want to import must be available through the network to the Media Server. You will need to know the hostname of the server containing the image(s) you want to upload and the hostname of the Media Server.
- Windows, Solaris, Linux, and VMware ESX operating system images on the Media Server must be available through `nfs/cifs/smb`.
- You must log in as an SA user (username and password) that has the required permissions to use the Import Media tool. If you do not specify a username/password in the `import_media` argument, you are prompted for a valid user name and password when you execute the command.

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 media hosted on an SA SMB server share:
`smb://<SMB Server>/OSMEDIA/<path>`
- Window media in CIFS server share:
`cifs://<CIFS Server>/<share>/<path>`

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.

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

Table 3 Import Media Tool Command Line Options

Import Media Tool Option	Description
<code>--help</code>	Display this help.
<code>--folder</code>	Override Folder location. Default is: <code>"/Package Repository/OS Media/<Platform Name>".</code>
<code>--medianame=<displayname></code>	Override automatically-generated display name. Note: Use <code>'_'</code> to escape spaces in the name.

Table 3 Import Media Tool Command Line Options (cont'd)

Import Media Tool Option	Description
<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 provisioning (default is no).

Configuring the Media Server for Microsoft Windows OS Media/Image

Perform the following tasks:

- 1 On the Media Server host, create the directory structure for the versions of the operating system that you plan to use for server provisioning. Ensure that you use the path names specified for the Media Server during SA installation.

Create the directory structure based on the root directories specified for the operating system media during SA installation. If necessary, contact your SA administrator for the locations of the operating system media root directories.

- 2 Ensure that the media for each operating system that you want to provision is available on the Media Server.
- 3 Copy the operating system media files to the location on the Media Server specified during the SA installation.

Importing Windows Media from Linux Host

When you launch the `import_media` tool from a server running a Red Hat Linux 5 kernel or higher, you must use the Import Media tool Windows CIFS syntax to import Windows media.

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

Importing Windows Media from a Solaris Host

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

Configuring the Media Server for Windows Server 2003 (x86/x86_64), 2008 and 2008 R2 x64 OS media

Perform the following tasks:

- 1 On the Media Server host, create the directory structure for the versions of the operating system that you plan to use for server provisioning. Ensure that you use the path names specified for the Media Server during SA installation.

Create the directory structure based on the root directories specified for the operating system media during SA installation. If necessary, contact your SA administrator for the locations of the operating system media root directories.

- 2 Ensure that the media for each operating system that you want to provision is available on the Media Server.
- 3 Copy the operating system media files to the location on the Media Server specified during the SA installation using the Import Media tool.

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:

```
$OEM$/$/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 using the Import Media tool.
- 6 If you need to specify OEM drivers, add a line similar to the following in the [Unattended] section of the unattend.txt file and reference the directory win which you are storing the drivers. For example:

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

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

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 hard coded 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%\m$lanman\\net")  
logonCmd.append("logon")  
logonCmd.append(self.mrl_username)  
logonCmd.append(self.mrl_password)
```

Configuring the Media Server for Red Hat Linux or VMware ESXi OS media

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

5 **Mount the first image read only:**

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

6 **Change to the temporary directory**

```
cd <tmp_mount_dir>
```

7 **Issue the command**

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

8 **cd out of the temporary directory.**

9 **Unmount the temporary directory:**

```
umount <tmp_mount_dir>
```

10 **Repeat steps from 5 to 9 for the remaining 4 images.**

11 **You can now import the media using the Import Media tool.**

Configuring the Media Server for SUSE Linux or SUSE Enterprise Linux OS media

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 Enterprise Server 10

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

Suse Linux Enterprise Server 11

You can install everything into a single directory, however, it is important that you copy the contents of the second Suse Linux Enterprise Server 11 DVD into the directory first, then copy the contents of the first Suse DVD into the same 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 Suse Linux Enterprise Server 9, 10 and 11 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

Configuring the Media Server for Sun Solaris

- 1 Download the Solaris 10 images to the Core.
- 2 Connect to the Core as root using ssh (you will need to run `mount` commands).
- 3 Create a temporary folder for loop mounting the images.
- 4 Create a directory under the media server's Linux media path. The Linux media path is a NFS share configured during the core install.
- 5 Mount the first image read only:

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

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

```
tar cf - . |(cd /media/hp/sunos/Solaris10/ && tar xfps -)
```
- 8 cd out of the temporary directory.
- 9 Unmount the temporary directory:

```
umount <tmp_mount_dir>
```
- 10 Repeat steps from 5 to 9 for the remaining 4 images.
- 11 You can now import the media using the Import Media tool.

Steps to Create MRLs

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

- 1 Log into the Software Repository (Slice Component bundle) host as root.
- 2 Change to the following directory:

```
/opt/opsware/mm_wordbot/util
```
- 3 Ensure that you have the correct path to the directory where you uploaded the operating system media on the OS Media Server.

Run the following `import_media` script:

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

For example, to import Windows Server 2003 operating system media from an SMB share named OSMEDIA on the server `mediasrv`, enter:

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


For example, to import Windows Server 2008 R2 x64 operating system media from and SMB share named OSMEDIA on the server mediasrv, enter:

```
mkdir <tmp_dir>
mount -t udf -o loop,ro w2k8r2sp1.iso <tmp_dir>
cd <tmp_dir> && tar cf - . | (cd /media/opsware/windows/w2k8sp1.r2 && tar xvf -)
import_media smb://mediasrv/OSMEDIA/w2k8sp1.r2
umount <tmp_dir>; rmdir <tmp_dir>
```

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 34.

Media Resource Locator (MRL) Administration

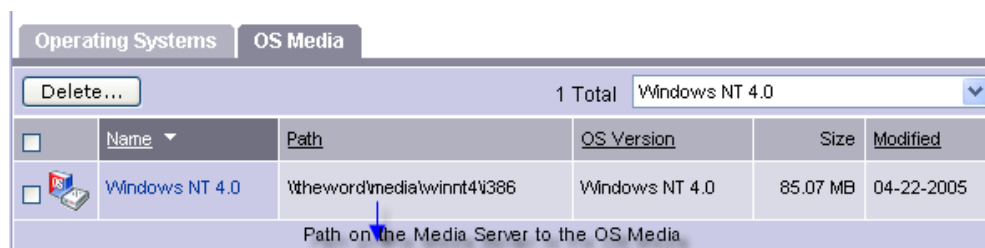
Editing MRLs

Perform the following steps to edit an MRL:

- 1 Log into the SA Web Client. The SA 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 5](#).

Figure 5 OS Media Page in the SA Web Client



- 4 Click the display name for the MRL that you want to edit. The **Edit OS Media** page appears, as [Figure 6](#) shows.

Figure 6 Edit OS Media Page in the SA Web Client

Name:	Red Hat Enterprise Linux AS 3
Description:	Red Hat Enterprise Linux AS 3 Media
OS Version:	Red Hat Enterprise Linux AS 3
Path:	nfs://mediaserver.c76.dev.opsware.com/media/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 MRLs

You cannot delete an MRL with the SA 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 and Managing OS Installation Profiles](#) on page 58 for more information.

Perform the following steps to delete an MRL:

- 1 Log into the SA Web Client. The SA 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 operating system 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.

Advanced Import Media Tool Information

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

An MRL is a network path (in URI format) to the installation media for an operating system on the Media Server. When a server is being provisioned with an operating system, the server mounts the network path for the operating system 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 operating system (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 provisioning. You can specify `--upload = yes` if you want to upload all packages to the Software Repository. The default is `--upload = no`.

The `--folder` option allows you to specify the full path to upload the operating system 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 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.

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

Configuring RAID on HP ProLiant Servers Before OS Provisioning

You can configure disk mirroring and striping as part of the initial setup of an HP ProLiant server prior to provisioning an operating system.

HP ProLiant RAID configuration requires having an HP ProLiant server configured with a *baseline RAID configuration* that is captured to a software policy. The captured RAID configuration is then applied to a server using the methods described in this section.

Supported Hardware

- HP ProLiant Servers

Supported Operating Systems

Baseline HP ProLiant RAID Configuration Capture

HP ProLiant RAID configuration capture is supported by the following SA-provided boot images:

- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 6 - OGFS-based (for additional information about capturing HP ProLiant RAID configuration using an OS Build Plan, see the `readme` file provided with the baseline Red Hat Enterprise Linux 6-based HP ProLiant RAID capture OS Build Plan)

- Microsoft Windows WinPE32, WinPE64, Winpe32-ogfs and Winpe64-ogfs (for additional information about capturing HP ProLiant RAID configurations using an OS Build Plan, see the `readme` file provided with the baseline Windows WinPE32 or WinPE64 HP ProLiant RAID capture OS Build Plans)



Solaris (SPARC, x86) is *not* supported.

HP ProLiant RAID Provisioning

- **Linux OS Sequences:** HP ProLiant RAID provisioning can be performed on any SA-supported Linux operating system that can be installed on HP ProLiant servers.
- **Windows OS Sequences:** HP ProLiant RAID provision can be performed on any SA-supported Windows version that can be installed on HP ProLiant servers.
- **Linux OS Build Plans:** HP ProLiant RAID provisioning is supported for
 - Red Hat Enterprise 5
 - Red Hat Enterprise 5 x64
 - Red Hat Enterprise 6
 - Red Hat Enterprise 6 x64

For additional information about provisioning HP ProLiant RAID configurations using OS Build Plans, see the `readme` file provided with the baseline Red Hat Enterprise Linux 6-based HP ProLiant RAID capture OS Build Plans.

- **Windows OS Build Plans:** HP ProLiant RAID provisioning is supported for
 - Windows 2003
 - Windows 2003 x64
 - Windows 2008
 - Windows 2008 x64
 - Windows 2008 R2 x64

For additional information about provisioning HP ProLiant RAID configurations using OS Build Plans, see the `readme` file provided with the baseline Windows WinPE32 or WinPE64 HP ProLiant RAID capture OS Build Plans.



The Red Hat Enterprise Linux 5/ Linux 6 boot images (Red Hat enterprise Linux 5.6 and 6.0 base) use a newer version of the Array Configuration Utility (ACU) tool. Therefore, HP ProLiant RAID configurations captured using the Red Hat Enterprise Linux 5 boot image can be successfully deployed only on unprovisioned servers that registered with the SA Core using the `linux5/linux6` boot images. Deployment of an HP ProLiant RAID configuration captured with the `linux5` (Red Hat Enterprise Linux 5 base) boot image to an unprovisioned server that registered with the SA Core using a different boot image will fail due to differing ACU tool versions

HP also occasionally updates the ACU tool which, in rare cases, can cause RAID configurations captured by an older version of the tool to be invalid. In these cases, you should rerun the RAID capture as described below in order to update the RAID capture.

Capture a Baseline HP ProLiant RAID Configuration

In order to configure RAID for an HP ProLiant server, you must first *capture* a baseline HP ProLiant RAID configuration that is saved into a RAID software policy that will be applied when provisioning new servers. SA uses the HP SmartStart Array Configuration Utility to perform the capture. The utility is installed by the SA installation.

To capture the RAID configuration, you must specify the custom attribute, `raid.capture=1` in the server record for the baseline HP RAID server which causes the server's RAID configuration to be captured into the software policy when it is booted into the SA Unprovisioned Server Pool.

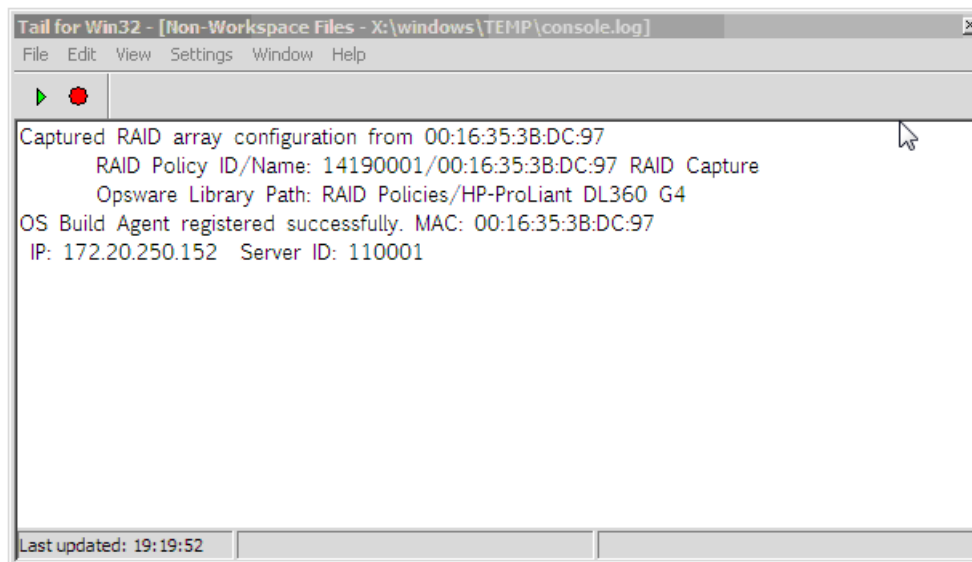
You can do this in either of two ways:

- a Use the Manage Boot Client (MBC) utility to create a server record for that server with the custom attribute `raid.capture=1` specified. See [The Manage Boot Clients \(MBC\) Option](#) on page 119 for information on creating or modifying a server record with MBC.
- b Reset the baseline HP RAID server to an SA Unprovisioned Server Pool to create the server record, edit the server record in the SA Client to specify the custom attribute `raid.capture=1`, then power the server off.

After the server record is created with the `raid.capture=1` custom attribute, boot the server into the SA Unprovisioned Server Pool so that the HP server's RAID configuration is captured in a software policy. Before SA creates the RAID software policy, it first creates a containing folder which is automatically named using the model number of the server for which the policy is to be created.

If the RAID configuration is captured successfully, you see a message similar to [Figure 7](#) in the console.log file:

Figure 7 RAID Configuration Capture Message

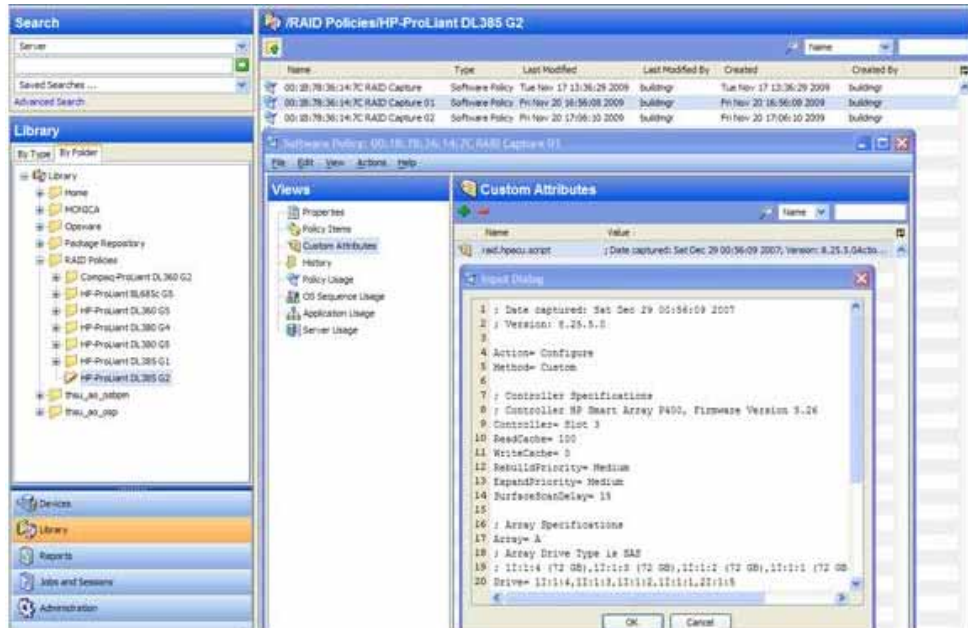


By default, the software policy is given a name that consists of the server's MAC address appended with the words RAID Capture, such as `00:16:35:3B:DC:97 RAID Capture`. You can rename the file in the SA Client. After the RAID configuration is captured, the value of the custom attribute `raid.capture` is automatically set to "0". This is to prevent unintended RAID captures from occurring for subsequent booting of the server to the unprovisioned servers pool.

The value of a custom attribute, `raid.version`, is also set to one of the following values: `linux`, `linux4`, `linux5`, or `winpe`. During an OS Sequence job, if the `raid.policy_id` is set, SA compares the `raid.version` value with the current server's version. If the values do not match, or the policy does not have the `raid.version` custom attribute, a warning is logged to the log file indicating the versions mismatch and that RAID deployment may fail. However, SA will attempt to continue the job.

The software policy appears in the SA Client RAID Policies Library:

Figure 8 RAID Policies Library in the SA Client



At this point, to provision RAID servers, you must add a server record custom attribute, `raid.policy_id=<value>` for the unprovisioned server, specifying the RAID software policy Object ID as the value. The captured baseline RAID configuration specified in the policy is then applied during provisioning.

The RAID policy you specify for an OS Sequence RAID deployment must be saved in the `/RAID Policies/Model Name` folder. If the RAID policies are saved or moved to a different folder, attempting an OS Sequence RAID deployment will fail with a Software Policy not found error.

▶ The method described above is the only way to apply RAID policies. RAID policies must not be attached to any objects, including unprovisioned servers, device groups, OS Sequences, and so on.

Creating an HP ProLiant RAID Dynamic Server Group

After you have captured a baseline HP ProLiant RAID software policy, you can add a custom attribute, `raid.policy_id=<value>` (specifying the RAID software policy Object ID as the value) to a Dynamic Device Group. Any unprovisioned server subsequently attached to that Device Group will have the HP ProLiant RAID configuration applied when it is provisioned.



Due to the way server records are inserted into Dynamic Server Groups, RAID capture may be skipped when the server is inheriting the RAID configuration. In this case, you should manually specify the RAID configuration policy in the server record. If the server is not yet in the SA Server Pool, you must reboot the server.

Manually Specifying an HP ProLiant RAID Configuration

You can write your own HP ProLiant RAID configuration file to be applied when a server is provisioned. To do so, specify the `raid.hpacu.script` custom attribute in the server record. You can specify a pre-written file for the script to use for configuration or open the editor in the server record and enter the RAID configuration manually.

4 Defining Installation Profiles, Build Plans and OS Sequences

OS Provisioning requires that you use certain configuration files that define how SA is to perform provisioning:

- [Installation Profiles](#)
- [OS Build Plans](#)
- [OS Sequences](#)

This chapter describes how to define and administer these files.

OS Installation Profile Requirements

This section discusses the following topics:

- [Overview on page 49](#)
- [Specifying Software for OS Provisioning on page 51](#)
- [Configuration Files on page 51](#)
- [Sun Solaris Installation Profile Requirements on page 51](#)
- [Red Hat Linux Installation Profile Requirements on page 52](#)
- [VMware ESX Installation Profile Requirements on page 52](#)
- [SUSE Linux Installation Profile Requirements on page 52](#)
- [Microsoft Windows Installation Profile Requirements on page 53](#)

Overview

You use OS Installation Profiles in conjunction with OS Sequences. Installation profiles are not required for OS Build Plans. Installation profiles specify which operating system is to be installed and how it is to be configured. You should create your installation profiles before creating OS Sequences since each sequence must be associated with an Installation Profile.

Before you create your Operating System Installation Profiles, you should have already set up OS Provisioning as described in the *SA Simple/Advanced Installation Guide* and in [OS Provisioning Requirements on page 19](#) and have created MRLs pointing to the operating system media using the Import Media tool as described in [Setting Up the Media Server on page 33](#).

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

The process of creating an Operating System Installation Profile includes:

- 1 Specifying properties for the operating system.
- 2 Specifying the location of the operating system media from which to perform an installation by selecting an MRL. (See [Editing MRLs](#) on page 41 for more information on editing MRLs.)
- 3 Uploading the following installation resources used during unattended installation:
 - A standard configuration file for the operating system. (See [Configuration Files](#) on page 51 for more information.)
 - A build customization script, which can modify the installation process at certain points. (See [Creating Build Customization Scripts](#) on page 71 for more information.)
 - *Microsoft Windows Only*: a Hardware Signature, which contains hardware specific information. ([Hardware Preparation](#) on page 25 for more information.)

Table 4 compares the installation resources across operating systems.

Table 4 Installation Resources for OS Installation Profiles

Installation Resource	SUSE	Windows Server 2003	Windows Server 2008	Solaris	Linux or VMware ESX
Configuration File	YAST profile autoinst.xml	unattend.txt	unattend.xml	Jumpstart profile	Kickstart/ Weasel profile
Build Customization Script	Optional executable file: bcs.tgz containing "run" script	Optional executable file: WinPE: bcs.zip containing "runphase.bat" script	Optional executable file: 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	Optional filename.txt	Not required	Not required



The configuration file that you upload for each operating system 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 operating system.

You can edit an OS Installation Profile later to add support for new hardware or to change the way the operating system is installed. See [Modifying Existing OS Installation Profiles](#) on page 65 in this chapter for more information.

Specifying Software for OS Provisioning

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 specifying SA Software Policies that add the desired packages in an OS Sequence.

Configuration Files

For OS Sequence-based provisioning, depending on the operating system being provisioned, the following configuration file must be specified in an OS Installation Profile:

- *Solaris*
JumpStart profile
- *Red Hat Linux*
Anaconda (Kickstart configuration file)
- *VMware ESX*
ESX 3.5: Anaconda (Kickstart configuration file)
ESX 4: Weasel (Kickstart configuration file)
- *SUSE Linux*
YaST2 configuration file
- *Windows*
unattend.txt or unattend.xml



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 Simple/Advanced Installation Guide* for details. To help isolate firewall related issues, you should leave firewalls disabled while configuring OS Provisioning for the first time and reenable them once the system is correctly configured. For Red Hat Enterprise Linux 5 and 6, the following line in your `ks.cfg` profile enables the firewall and allows the SA Agent to function correctly:

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

For VMware ESX prior to 4.1 provisioning, the SA Agent installer may *temporarily* bypass any OS-based firewall configured in the `ks.cfg`.

Sun Solaris Installation Profile Requirements

When preparing a Solaris OS Installation Profile, OS Provisioning requires that you upload a JumpStart profile.

The Solaris Jumpstart file must:

- Be a valid profile that you can 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 Installation Profile Requirements

The Red Hat Linux Kickstart file specifies 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 Kickstart file must:

- Be a valid configuration file that you can use with a Kickstart server.
- Specify the *RPM packages* to install.
- Include the *reboot option*.

VMware ESX Installation Profile Requirements

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 file must:

- Be a valid configuration file that you can 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 Installation Profile Requirements

The SUSE Linux configuration file specifies to YaST2 which packages to install, how to partition the drive, and the operating system 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 YaST2 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 Installation Profile Requirements

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 operating system 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 Server 2003](#) below and [Sample Response File for Windows Server 2008 x86](#) on page 54 and [Sample Response File for Windows Server 2008 x64](#) on page 56 for examples of valid Windows response files.

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 SA 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

```

Sample Response File for Windows Server 2008 x86

The following sample response file shows typical valid responses for a Windows Server 2008 installation using an OS Build Plan. This sample response file contains the required settings for Windows Server 2008 provisioning with SA OS Provisioning. Certain custom attributes listed may not be supported by OS Sequences.

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Unattend.xml file tested on the following setups:
HP Server Automation 9.00.00
VMware ESX Server 4.0 with Virtual Machine version 7 guests
guests configured for Windows 2008 x86 guest OS, 512 MB RAM, 15 GB disk, E1000
NIC
-->
<unattend xmlns="urn:schemas-microsoft-com:unattend" xmlns:wcm="http://
schemas.microsoft.com/WMIConfig/2002/State">
    <settings pass="windowsPE">
        <component name="Microsoft-Windows-International-Core-WinPE"
processorArchitecture="x86" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
            <SetupUILanguage>
                <UILanguage>en-US</UILanguage>
            </SetupUILanguage>
            <InputLocale>en-US</InputLocale>
            <SystemLocale>en-US</SystemLocale>
            <UILanguage>en-US</UILanguage>
            <UILanguageFallback>en-US</UILanguageFallback>
            <UserLocale>en-US</UserLocale>
        </component>
        <component name="Microsoft-Windows-Setup" processorArchitecture="x86"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <UserData>
                <ProductKey>
                    <WillShowUI>OnError</WillShowUI>
                    <Key>@ProductKey@</Key>
                </ProductKey>
                <AcceptEula>true</AcceptEula>
            </UserData>
            <DiskConfiguration>

```

```

    <Disk wcm:action="add">
      <CreatePartitions>
        <CreatePartition wcm:action="add">
          <Extend>>true</Extend>
          <Order>1</Order>
          <Type>Primary</Type>
        </CreatePartition>
      </CreatePartitions>
      <ModifyPartitions>
        <ModifyPartition wcm:action="add">
          <Active>>true</Active>
          <Format>NTFS</Format>
          <Label>OS</Label>
          <Letter>@SystemDrive:C@</Letter>
          <Order>1</Order>
          <PartitionID>1</PartitionID>
        </ModifyPartition>
      </ModifyPartitions>
      <DiskID>@SystemDiskNumber:0@</DiskID>
      <WillWipeDisk>>true</WillWipeDisk>
    </Disk>
    <WillShowUI>Never</WillShowUI>
  </DiskConfiguration>
  <ImageInstall>
    <OSImage>
      <InstallFrom>
        <MetaData wcm:action="add">
          <Key>/IMAGE/Name</Key>
          <Value>@WindowsImageName:Windows Longhorn
SERVERENTERPRISE@</Value>
        </MetaData>
      </InstallFrom>
      <WillShowUI>Never</WillShowUI>
      <InstallTo>
        <DiskID>@SystemDiskNumber:0@</DiskID>
        <PartitionID>1</PartitionID>
      </InstallTo>
    </OSImage>
  </ImageInstall>
</component>
</settings>
<settings pass="oobeSystem">
  <component name="Microsoft-Windows-Shell-Setup"
processorArchitecture="x86" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
    <UserAccounts>
      <AdministratorPassword>
        <Value>@AdminPassword:ChangeMe123!@</Value>
        <PlainText>>true</PlainText>
      </AdministratorPassword>
    </UserAccounts>
    <OOBE>
      <HideEULAPage>>true</HideEULAPage>
    </OOBE>

```

```

        <RegisteredOrganization>HP</RegisteredOrganization>
        <RegisteredOwner>SA</RegisteredOwner>
    </component>
    <component name="Microsoft-Windows-International-Core"
processorArchitecture="x86" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
        <InputLocale>en-US</InputLocale>
        <SystemLocale>en-US</SystemLocale>
        <UILanguage>en-US</UILanguage>
        <UILanguageFallback>en-US</UILanguageFallback>
        <UserLocale>en-US</UserLocale>
    </component>
</settings>
<settings pass="specialize">
    <component name="Microsoft-Windows-Shell-Setup"
processorArchitecture="x86" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
        <ComputerName>@ComputerName: *@</ComputerName>
    </component>
</settings>
<cpi:offlineImage cpi:source="wim:d:/sources/
install.wim#@WindowsImageName:Windows Longhorn SERVERENTERPRISE@"
xmlns:cpi="urn:schemas-microsoft-com:cpi" />
</unattend>

```

Sample Response File for Windows Server 2008 x64

The following sample response file shows typical valid responses for a Windows Server 2008 R2 installation using an OS Build Plan. This sample response file contains the required settings for Windows Server 2008 x64 provisioning with SA OS Provisioning. Certain custom attributes listed may not be supported by OS Sequences.

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Unattend.xml file tested on the following setups:
HP Server Automation 9.00.00
VMware ESX Server 4.0 with Virtual Machine version 7 guests
guests configured for Windows 2008 x86 guest OS, 512 MB RAM, 15 GB disk, E1000
NIC
-->
<unattend xmlns="urn:schemas-microsoft-com:unattend" xmlns:wcm="http://
schemas.microsoft.com/WMIconfig/2002/State">
    <settings pass="windowsPE">
        <component name="Microsoft-Windows-International-Core-WinPE"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
            <SetupUILanguage>
                <UILanguage>en-US</UILanguage>
            </SetupUILanguage>
            <InputLocale>en-US</InputLocale>
            <SystemLocale>en-US</SystemLocale>
            <UILanguage>en-US</UILanguage>

```



```

        <UILanguageFallback>en-US</UILanguageFallback>
        <UserLocale>en-US</UserLocale>
    </component>
    <component name="Microsoft-Windows-Setup"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
        <UserData>
            <ProductKey>
                <WillShowUI>OnError</WillShowUI>
                <Key>@ProductKey@</Key>
            </ProductKey>
            <AcceptEula>true</AcceptEula>
        </UserData>
        <DiskConfiguration>
            <Disk wcm:action="add">
                <CreatePartitions>
                    <CreatePartition wcm:action="add">
                        <Extend>true</Extend>
                        <Order>1</Order>
                        <Type>Primary</Type>
                    </CreatePartition>
                </CreatePartitions>
                <ModifyPartitions>
                    <ModifyPartition wcm:action="add">
                        <Active>true</Active>
                        <Format>NTFS</Format>
                        <Label>OS</Label>
                        <Letter>@SystemDrive:C@</Letter>
                        <Order>1</Order>
                        <PartitionID>1</PartitionID>
                    </ModifyPartition>
                </ModifyPartitions>
                <DiskID>@SystemDiskNumber:0@</DiskID>
                <WillWipeDisk>true</WillWipeDisk>
            </Disk>
            <WillShowUI>Never</WillShowUI>
        </DiskConfiguration>
        <ImageInstall>
            <OSImage>
                <InstallFrom>
                    <MetaData wcm:action="add">
                        <Key>/IMAGE/Name</Key>
                        <Value>@WindowsImageName:Windows Longhorn
SERVERENTERPRISE@</Value>
                    </MetaData>
                </InstallFrom>
                <WillShowUI>Never</WillShowUI>
                <InstallTo>
                    <DiskID>@SystemDiskNumber:0@</DiskID>
                    <PartitionID>1</PartitionID>
                </InstallTo>
            </OSImage>
        </ImageInstall>
    </component>

```

```

    </settings>
    <settings pass="oobeSystem">
      <component name="Microsoft-Windows-Shell-Setup"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
        <UserAccounts>
          <AdministratorPassword>
            <Value>@AdminPassword:ChangeMe123!@</Value>
            <PlainText>>true</PlainText>
          </AdministratorPassword>
        </UserAccounts>
        <OOBE>
          <HideEULAPage>>true</HideEULAPage>
        </OOBE>
        <RegisteredOrganization>HP</RegisteredOrganization>
        <RegisteredOwner>SA</RegisteredOwner>
      </component>
      <component name="Microsoft-Windows-International-Core"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
        <InputLocale>en-US</InputLocale>
        <SystemLocale>en-US</SystemLocale>
        <UILanguage>en-US</UILanguage>
        <UILanguageFallback>en-US</UILanguageFallback>
        <UserLocale>en-US</UserLocale>
      </component>
    </settings>
    <settings pass="specialize">
      <component name="Microsoft-Windows-Shell-Setup"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
        <ComputerName>@ComputerName: *@</ComputerName>
      </component>
    </settings>
    <cpi:offlineImage cpi:source="wim:d:/sources/
install.wim#@WindowsImageName:Windows Longhorn SERVERENTERPRISE@"
xmlns:cpi="urn:schemas-microsoft-com:cpi" />
  </unattend>

```

Defining and Managing OS Installation Profiles

This section discusses the following topics:

- [Defining an OS Installation Profile — Linux/Unix on page 59](#)
- [Defining an OS Installation Profile — Windows on page 60](#)
- [Modifying Existing OS Installation Profiles on page 65](#)
- [Changing the OS Installation Profile Properties on page 65](#)

- [Modifying How an OS Is Installed on a Server — Linux/Unix](#) on page 66
- [Modifying the OS Installation Profile Packages](#) on page 68
- [Viewing Change History for an OS Installation Profile](#) on page 69
- [Deleting an OS Installation Profile](#) on page 70

Defining an OS Installation Profile — Linux/Unix

To use the Prepare Operating System Wizard to define a Linux/Unix OS Installation Profile, perform the following steps:

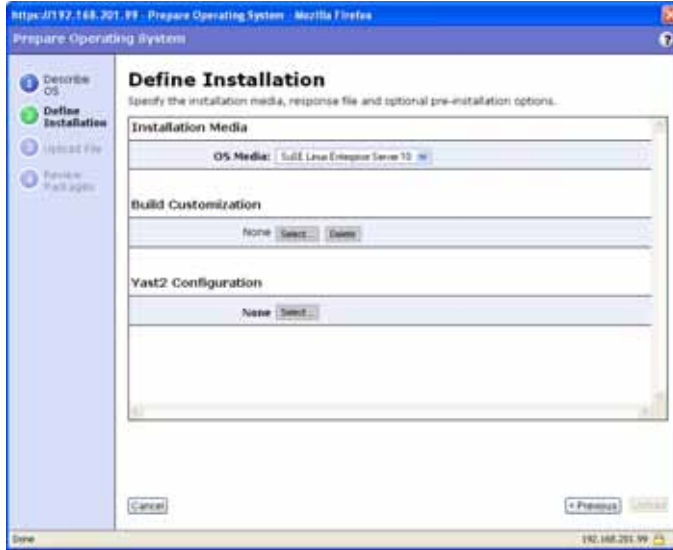
- 1 Access the *Prepare Operating System* wizard from the SA Client or the SAS Web Client:
 - *SA Client*: from the **Navigation** pane, select **Library** ► **OS Installation Profiles**. Select an operating system, then from the **Actions** menu, select **Create New**.
 - *SA Web Client Home Page*: click **Prepare OS** in the **Tasks** panel or, from the **Navigation** pane, click **Software** ► **Operating Systems**. The **Operating Systems** page appears. Click **Prepare OS**.

The *Describe OS* page appears, as [Figure 9](#) shows.

Figure 9 Describe OS Page in the Prepare Operating System Wizard

- 2 Describe the operating system by specifying the following information:
 - **Name:** (*Required*) Specify the display name for the Linux/Unix operating system.
 - **Description:** (*Optional*) Provide a text description to identify the platform and hardware support.
 - **Customer:** (*Required*) Associate the Linux/Unix operating system with a specific customer; to set up the operating system for use by all customers, select “Customer Independent”.
 - **OS Version:** (*Required*) Specify the version of the Linux/Unix operating system (select from a pre-defined list of operating systems that SA supports).
- 3 Click **Next**. The *Define Installation* page is displayed, as [Figure 10](#) shows.

Figure 10 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 Linux/Unix operating system (select one MRL from the pre-defined drop-down list of available MRLs).
See [Creating Media Resource Locators \(MRLs\)](#) on page 33 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 SA Web Client, see [Using Build Customization Scripts](#) on page 71.)
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 Linux/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 operating system preparation process.
- 6 Click **Close** when the upload is completed.

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 Access the *Prepare Operating System* wizard from the SA Client or the SAS Web Client:
 - SA Client: from the **Navigation** pane, select **Library** ► **OS Installation Profiles**. Right click on an operating system and select **New**.

- *SA 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**.

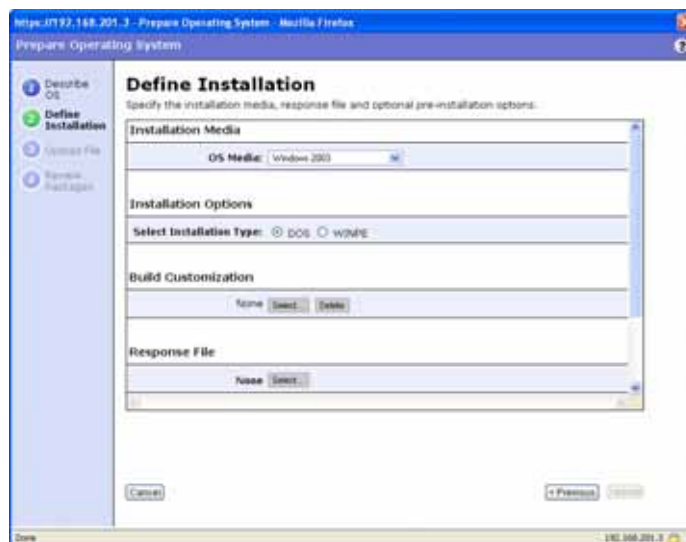
The *Describe OS* page appears, see [Figure 11](#).

Figure 11 Prepare OS Wizard: Describe OS



- 2 Describe the operating system by specifying the following information:
 - **Name:** (*Required*) Specify the display name for the Windows operating system.
 - **Description:** (*Optional*) Provide a text description to identify the platform and hardware support.
 - **Customer:** (*Required*) Associate the Windows operating system with a specific customer; to set up the operating system for use by all customers, select “Customer Independent”.
 - **OS Version:** (*Required*) Specify the version of the Windows operating system (selected from the pre-defined list of the operating systems that SA supports).
- 3 Click **Next**. The *Define Installation* page appears, see [Figure 12](#).

Figure 12 Prepare OS Wizard: Define Installation



- 4 Define the installation by providing the following information:
 - **OS Media:** (*Required*) Specify the MRL for the Windows operating system (select one MRL from the pre-defined drop-down list of available MRLs that you have already defined). See [Creating Media Resource Locators \(MRLs\)](#) on page 33 for more information on this topic.
 - **Installation Options:** (*Required*) Choose the type of pre-installation environment to use when you install the Windows operating system.
Your selection determines which customization script options you can use.



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 operating system installation. Refer to the Microsoft Windows product documentation for more information.
- **Custom Disk Formatting:** This script is executed directly onto the hard drive during operating system installation.
- **Install Drive:** Indicates the drive letter on which to install the Windows operating system.

If you do not enter any settings in these fields, the default values used are shown in [Figure 13](#).

Figure 13 Default Values used for WinPE Installation Options in OS Installation Profile

Installation Options	
Select Installation Type:	<input type="radio"/> DOS <input checked="" type="radio"/> WINPE
Custom Disk Partitioning:	<pre>rescan select disk 0 clean create partition primary active assign letter=C</pre>
Custom Disk Formatting:	<pre>format.com C: /FS:NTFS /Q /Y /V:</pre>
Install Drive:	<input type="text" value="C"/>

- **Build Customization:** (*Optional*) Select a build script to customize the way the build process operates for the Windows operating system.

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 SA Web Client.

Click **Select** to choose a file from the popup window.

See [Creating Build Customization Scripts](#) on page 71 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 operating system 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 operating system preparation process.

- 6 Click **Close** when the process is complete.

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
```

```

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

;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=<org_name>
  ;OrgName=<org_name>
  ;You can/should also set this as a custom attribute
  ;"ProductKey"
  ;on the OS Installation Profile
  ProductKey=<product_key>

```



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.

Modifying Existing OS Installation Profiles

You can edit an OS Installation Profile by:

- Changing the properties for the operating system, for example which customer(s) can use the OS Installation Profile to provision servers.
- Modifying the way that the operating system 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 SA Web Client or from the SA Client. (See [Defining Custom Attributes](#) on page 83. For information on how to set custom attributes for software policies, see [Adding Custom Attributes to OS Installation Profile \(SA Web Client\)](#) on page 89.
- 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 operating system that you want to edit. The **Edit Operating System** page appears.
- 3 Select the **Properties** tab (see [Figure 14](#)). You can modify the following settings:
 - **Name:** Sets the display name for the operating system.
 - **Description:** Provides a text description of the operating system.
 - **Customer:** Associates the operating system 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 14 Properties Tab for an OS Installation Profile in the SA 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:	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 — Linux/Unix

To modify the way an operating system is installed on Linux/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 Linux/Unix operating system that you want to edit. The **Edit Operating System** page appears.
- 3 Select the **Installation** tab.
- 4 Modify the following settings:
 - **Installation Media:** (*Required*) Modify the MRL for the Linux/Unix operating system (select one MRL from the pre-populated drop-down list).
See [Creating Media Resource Locators \(MRLs\)](#) on page 33 for more information on this topic.
 - **Build Customization Script:** (*Optional*) Customize the way the build process operates for that Linux/Unix operating system (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 SA Web Client.
See [Creating Build Customization Scripts](#) on page 71 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 Operating System Is Installed on a Server — Windows

Perform the following steps to modify the way an operating system is installed on Windows 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 operating system 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 15](#).

Figure 15 Installation Tab for an OS Installation Profile in the SA 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:** Modify the MRL for the Windows operating system. Click **Select** and select an operating system media from the list in the popup window.
- **Installation Options:** If you selected WINPE when you created the Windows installation profile, you can modify 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 operating system 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 operating system 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 operating system on. The default drive letter used is C.

- **Build Customization Script:** Customizes the way the build process operates for that operating system. 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 SA 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 operating system 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 9.10, 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 [Defining and Managing OS Installation Profiles](#) on page 58 and [Creating OS Sequences](#) on page 100 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 operating system 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 16](#) shows.

Figure 16 Packages Tab for an OS Installation Profile in the SA 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 operating system.
- 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 SA Web Client and in the SA Client.

To view the history of changes to an OS Installation Profile in the SA 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 operating system 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 17](#).

Figure 17 History Tab for an OS Installation Profile in the SA 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 SA 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 operating system 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.

Creating Build Customization Scripts

This section discusses the following topics:

- [Using Build Customization Scripts](#) on page 71
- [Solaris Build Customization Scripts](#) on page 72
- [Linux Build Customization Scripts](#) on page 76
- [Windows Build Customization Scripts](#) on page 79

Using Build Customization Scripts

You can use operating system-specific build scripts to control the way each operating system is provisioned. Build scripts allow you to manage each operating system installation from the network connection to SA Agent installation.

OS Provisioning build scripts provide hooks into the build process that allow you to modify operating system installations at specific points. These hooks call a single build customization script at the appropriate time in the operating system installation process.

Because each build script is specific to the operating system it installs, build customization and installation vary by operating system. 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 operating system, 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 operating system 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 — Linux/Unix](#) on page 59 or [Defining an OS Installation Profile — Windows](#) on page 60 for more information.

Solaris Build Customization Scripts

This section describes creating build customization scripts for Sun Solaris.

The Sun Solaris Build Process

It is important to understand the Solaris build process before you include a build customization script for a Solaris installation profile. [Table 5](#) details the exact steps that occur when you provision an installation client with Solaris.

A user initiates the build process with Steps 1 and 5. The rest of the build process steps occur automatically in OS Provisioning.

Table 5 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 SA Web Client.

Table 5 Sun Solaris Build Process (cont'd)

Phase	Build Process Steps
Phase One	<p>5 In the SA Web Client, a user chooses to install an operating system 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 command 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 operating system 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 User Guide: Audit and Compliance* for more information on how remediation installs software on servers.

Requirements for Solaris Build Customization Scripts

Build customization script for Solaris 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.

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

If you must provision a Solaris server and the Boot Server is hosted on a Red Hat Enterprise Linux or Suse Linux Enterprise 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/nfsserver stop
/etc/init.d/nfsserver start
```

Creating a 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 operating system 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 SA Web Client output details for the operating system.

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 filesystems > /tmp/profile.$$
    echo "partitioning default" >> /tmp/profile.$$
    echo "filesystems 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 Customization Scripts

A 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.

Linux/Itanium Build Process

It is important to understand the Linux/Itanium build process before you include a build customization script in a Linux/Itanium OS Installation Profile. [Table 6](#) describes the exact steps that occur when you provision an installation client with Red Hat Linux, Red Hat Linux Itanium 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 Red Hat Linux Itanium and VMware ESX follows the same process as the Linux build process.

Table 6 Linux Build Process

Phase	Build Process Steps
Pre-installation	<ol style="list-style-type: none">1 A user boots the installation client from PXE or the Linux Boot CD ROM.2 The installation client loads a standard Red Hat boot image and mounts the second stage image specified by the kernel parameters. Note: During a PXE boot, the Build Agent is called from the kickstart file. When a CD install is specified, Anaconda is replaced by a custom SA script that is used to invoke the OS Build Agent.3 The OS Build Agent registers with the Build Manager.4 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 SA Web Client.
Phase One	<ol style="list-style-type: none">5 In the SA Web Client, a user selects the target version of Linux to install on the installation client.6 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.7 The Linux build script copies GRUB or eLilo onto the partition and installs it into the MBR.8 The Linux build script configures GRUB or eLilo to boot this partition, and kernel arguments are set to do an NFS installation on the location indicated by the MRL.9 If the Custom Attribute <code>kernel_arguments</code> is set for the OS Installation Profile, these kernel arguments are appended.10 The OS Build Agent exits and the server reboots.

Table 6 Linux Build Process (cont'd)

Phase	Build Process Steps
Phase Two	<p>11 The target boot image loads and runs the OS Build Agent.</p> <p>12 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>13 The Linux build script writes the configuration file defined by the MRL to the disk.</p> <p>14 If it exists, the Linux build script runs the build customization script.</p> <p>15 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 SA Client.</p> <p>16 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>17 When the operating system installation completes, Anaconda reboots the installation client, which boots from the newly installed operating system.</p>
Phase Three	<p>18 On entering multi-user mode, the OS Build Agent is invoked and contacts the OS Build Manager.</p> <p>19 The Linux build script installs the SA Agent.</p> <p>20 The Linux build script exits.</p> <p>The operating system installation section of provisioning is complete.</p>

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/Itanium Build Process](#) on page 76.

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 Customization Scripts

This section describes creating build customization scripts for Microsoft Windows.

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 7 details the steps that occur when you provision an installation client with Windows WinPE.

A user initiates the build process with Steps 1 and 6. The rest of the build process steps happen automatically in OS Provisioning.

Table 7 Microsoft Windows Build Process (WinPE)

Phase	Build Process Steps
Pre-installation	<p>1 A user boots an installation client over the network by using a PXE network bootstrap program or by using the WinPE.</p> <p>2 The user can install either WinPE x86 32 bit or WinPE x64 64 bit pre-installation environment.</p> <p>3 PXE boots the Windows OS Build Agent over the network. When using the WinPE pre-installation environment, you will not be prompted to create a disk partition.</p> <p>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 SA Web Client.</p>
Phase One	<p>5 The user selects a Windows server from the Server Pool list in the SA Web Client and assigns a Windows OS Installation Profile or a Windows template to the server.</p> <p>6 The Windows build script mounts the Windows installation media as indicated by the Media Resource Location (MRL).</p> <p>7 The Windows build script initiates a Windows unattended setup.</p> <p>8 The Windows build script waits for a Windows unattended setup to complete and Windows to boot for the first time.</p>
Phase Two	<p>9 Windows boots for the first time.</p> <p>10 If a build customization script was specified in the OS Installation Profile, it is executed by the Windows build script.</p> <p>11 The Windows build script installs the Agent. The Windows build script exits and Phase Two is complete.</p>

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.

Creating a Windows Build Customization Script (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
```

Defining Custom Attributes

This section discusses the following topics:

- [Custom Attributes for Sun Solaris on page 84](#)
- [Custom Attributes for Linux or VMware ESX on page 85](#)
- [Custom Attributes for Microsoft Windows on page 87](#)
- [Adding Custom Attributes to OS Installation Profile \(SA Web Client\) on page 89](#)
- [Adding Custom Attributes to OS Installation Profile \(SA Client\) on page 90](#)

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

The SA 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 \(SA Web Client\) on page 89](#).

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 \(SA Web Client\)](#) and [Adding Custom Attributes to OS Installation Profile \(SA Client\)](#) on page 90 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 SA Web Client.

The custom attributes correspond to the equivalent keywords in the `sysidcfg` file. See [Table 8](#).

Table 8 Sun Solaris Custom Attributes

Keyword	Description
<code>archive_location</code>	<p>NFS path to a Flash Archive (<code>flar</code>) to use instead of operating system media.</p> <p>Example Value: <code>nfs://mediaserver.company.com/flars/sunos5.10_basic.flar</code></p>
<code>boot_options</code>	<p>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>.</p> <p>Example Value: Values will vary, see your Solaris documentation.</p>
<code>reboot_command</code>	<p>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.</p> <p>Example Value: <code>net2:dhcp - install</code></p>
<code>root_password</code>	<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>
<code>timezone</code>	<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 <code>US/Pacific</code>. 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>

Table 8 Sun Solaris Custom Attributes (cont'd)

Keyword	Description
system_locale	<p>Sets the language for the configuration of the installation client (sets LANG in /etc/default/init). Valid locale values are installed in /usr/lib/locale. 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 /etc/default/nfs next to "NFSMAPID_DOMAIN=.</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>

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 operating system 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 operating system from CD-ROM or DVD).



Although custom attributes are provided with a default value, you must ensure that the values are valid for your system before proceeding.

Table 9 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: "rhe130", "rhel40", "rhel50", "rehl60", "rhel3ia", "rhel4ia", "rhel5ia" Note: This custom attribute is used only for <i>reprovisioning</i> . The value of this custom attribute specifies the type of kernel the server boots to during reprovisioning.
hpsa_netconfig	Created after using non-DHCP to boot the target server into the Unprovisioned Servers list. See page 108 .
kernel_arguments	Values: "noapci", "root=LABEL=/", "quiet", "splash"
ksdevice (Linux pxe boot)	Values: MAC address of the NIC 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: kernel_arguments =ksdevice=mac address ksdevice mac address When powering on this device and PXE booting it, you do not need to specify the kickstart device.
ksdevice (linux5, linux6 or linux6-ogfs)	Values: bootif Default: ksdevice=bootif for all Linux PXE types (including linux5, linux6, linux6-ogfs), to prevent prompting for the Kickstart device when booting a multiple NIC server into the Unprovisioned Server pool.

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 9](#) 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 Opsware
#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 operating system custom attributes that allow you to replace or insert values inside the `unattend.txt` file during the operating system 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 \(SA Web Client\)](#) on page 89 or [Adding Custom Attributes to OS Installation Profile \(SA Client\)](#) on page 90.

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 10 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.
hpsa_netconfig	None	Created after using non-DHCP to boot the target server into the Unprovisioned Servers list. See page 115 .
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.

Table 10 Windows Custom Attributes for OS Provisioning (cont'd)

Keyword	Corresponding unattend.txt Attribute	Description
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.
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 operating system installation fails with a timeout error.



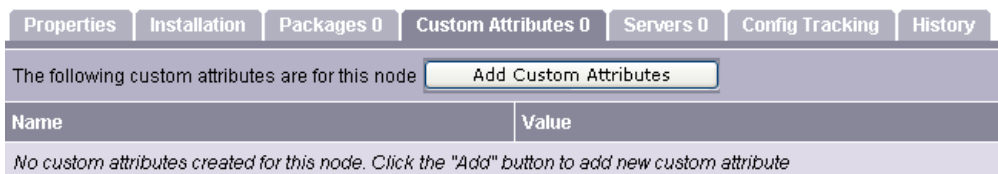
The Windows OS Provisioning custom attribute `argstring` is not supported with OS Build Plans.

Adding Custom Attributes to OS Installation Profile (SA Web Client)

Perform the following steps to add custom attributes to an OS Installation Profile in the SA Web Client:

- 1 From the **Navigation** pane inside the SA Web Client, click **Software ► Operating Systems**. The **Operating Systems** page appears.
- 2 Click the name of the operating system 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 18](#) shows.

Figure 18 The Custom Attributes Tab for an OS Installation Profile in the SA 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:
 - *SA Web Client home page*: From the Power Tools section
 - *SA Web Client Menu*: From **Start** ► **All Programs** ► **SA Client**
- 2 From inside the SA 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**.

Creating OS Build Plans

As a replacement for OS Sequences, SA provides a new, more flexible method for Windows, Red Hat Enterprise Linux, and VMware ESXi hosts to specify how an operating system is installed called *OS Build Plans*. You use an OS Build Plan to specify server provisioning details, such as operating system configuration information, software, customization scripts and patch policies. While similar to the OS Sequence capability, OS Build Plans provide these functional improvements over OS Sequences:

- As of SA 9.1, baseline OS Build Plans are installed as part of the SA Core installation or upgrade. To use these baseline OS Build Plans you can open a baseline OS Build Plan in the SA Client Library and use the Copy Plan button to create a customized Build Plan. In most cases, you will need only to specify the Media Server IP address/FQDN and the media path for your system.
- The order of remediation, or run server scripts, are clearly defined in the OS Build Plan wizard. OS Build Plans execute the scripts in the order listed.
- OS Build Plans make it easier to customize the operating system installation to meet your specific needs, for example:
 - Integration with other internal systems at specific points during the operating system build phase.
 - Running a RAID configuration utility or a firmware update

- Modifying the unattend.xml or kickstart file from a script before beginning an installation process
- Simpler architecture. OS Build Plans use the same network ports and protocols as a full SA Agent. Fewer SA Core Components are involved.
- OS Build Plans use the more robust and powerful execution environment of the Global Shell (OGFS).
- A more transparent build process means easier progress monitoring and troubleshooting.
- The use of an OGFS Agent provides an easy way to configure and troubleshoot servers before or during an operating system build.
- OS Build Plans allow simpler set up:
 - Running the `import_media` utility is no longer required.
 - Defining OS Installation Profiles in the SA Web Client is now optional, not required.
- No separate client installation is required to deploy operating systems.
 - The new Run OS Build Plan wizard is a web application.
 - The SA Client can be used to define OS Build Plans.
 - Build Plan APXs can be run from the command line or from scripts.
- Perform other tasks beyond OS Installation. For example, OS Build Plans can be created for image capture, file restore, or secure data erasure.


 You can still use OS Sequences to configure your Windows or Linux operating system installation. The functionality is still fully available. See [Creating OS Sequences](#) on page 100. HP recommends, however, that you explore the advanced features of OS Build Plans and consider migrating from OS Sequences to OS Build Plans. AS of SA 9.1, OS Build Plans are available for Windows Server 2003, 2008, 2008 R2 x64, Red Hat Enterprise Linux Server 5 and 6 and VMware ESXi 4.1 system installations.

Support for the OGFS Agent

SA provides service OS boot images that contains Linux (x86) or Windows (x86, x86_64) versions of the OGFS Agent. Servers can be booted into the OGFS Agent which enables temporary Global Shell access called "Maintenance Mode" as well as execution of OS Build Plans.

The Linux and Windows OGFS Agent is available through PXE and CD-ROM boot. There are new options on the PXE boot menu labeled `linux6-ogfs`, `winpe32-ogfs`, and `winpe64-ogfs` that enable server maintenance mode and OS Build Plans for Linux and Windows-based systems. The bootable CD-ROM image is available for export from the SA Library.

The images include both the OS Build Agent required to run OS Sequences and the OGFS Agent required to run OS Build Plans.

 For Windows, SA provides specific images for booting when using an OS Sequence and for booting using an OS Build Plan. For Linux, a single boot image handles both environments.

A menu displayed during the boot process allows the user to select which agent to start.

When booted with this OGFS Agent image, servers are available in the Global Shell. They appear in the SA Client identified by a Maintenance Mode icon. When a server is in *Maintenance state* it is running the OGFS Agent and is accessible using Global Shell commands. Inside the Global Shell, you can execute commands on the target servers using the `rosh` utility and browse their filesystems:

Table 11 File System Access Levels

File System	Access Level
Ext2/3/4 (Linux)	Read/Write
ReiserFS (Linux)	Read Only
NTFS (Windows)	Read Only
UFS (Solaris)	Read Only
ZFS (Solaris)	Read Only

You can boot new bare metal servers, unprovisioned servers, and existing managed servers into the OGFS Agent.



When booting existing managed servers into maintenance mode, the following caveats apply. In order to securely register with SA and match up to the correct SA device record, the OGFS Agent must locate and access the managed server's unique ID and cryptographic certificate from one of the existing filesystems. In most cases this happens automatically and the server is able to successfully register as an existing managed server booted into maintenance mode. However, in more complex filesystem layouts, the necessary cryptographic certificate may not be able to be located. In this case, the OGFS Agent cannot register with the SA core. A message is displayed on the console indicating in order to force registration with the core, the server must be deactivated in SA.

In SA 9.10, the Linux OGFS Agent is provided. Therefore you can run OS Build Plan that have Unix server scripts and/or all other supported object types steps against the Linux OGFS Agent.

Effect of the OGFS Agent on Server Lifecycle

- Servers running the OGFS Agent are ready for OS Build Plans or ad-hoc OGSHELL access.
- When an OS Build Plan is running, the server's lifecycle is set to `PROVISIONING`. This prevents running more than one job against the same server simultaneously.
- When an OS Build Plan completes, the final state of the target server is affected by several conditions as well as the server's initial lifecycle when the OS Build Plan was first launched.

What are OS Build Plans?

When you install (provision) an operating system on a server, you may have certain configurations settings you want to specify for all similar servers, you may also want to set environment variables, install and configure applications, configure RAID settings, and so on.

SA OS Build Plans provide a framework you can use to design server installation templates that configure a server exactly as you want it configured during OS Provisioning.

OS Build Plans take advantage of the SA Global File System (OGFS) and Automation Platform Extensions (APXs) to install operating systems on unprovisioned servers.

Using SA OS Build Plans, you can specify the following tasks to be performed during OS Provisioning:

Run scripts

You can create and specify OGFS scripts and server scripts (with arguments) for answer file processing, custom attribute handling, customer assignments, and so on and server scripts for reboots and general functionality to be run during OS Provisioning.

- **Install zip packages**

You can specify zip packages to be installed which can be used to deliver any additional software utilities you may need during the OS installation. For example, you might need to install some additional driver software or make use of a third party image installation program.



OS Build Plans do not process pre/post scripts nor reboot settings on ZIP packages. Such settings apply only to Software Policy usage of ZIP packages. Pre/post scripts can be defined as separate Build Plan steps. Rebooting should be handled by the boot script provided for this purpose.

- **Attach Patch Policies**

You can specify the Patch Policies that must be applied.

- **Attach Software Policies**

You can specify the Software Policies that must be applied

- **Server remediation**

You specify the remediation's reboot options and Error Handling option.

- **Add server to Static Device Groups**

You can specify a static Device Group to which the server should be added.

Baseline OS Build Plans

SA provides a set of baseline OS Build Plans that you copy and use to base your Build Plans on. These Build Plans are installed by default during SA installation or upgrade. Each build plan is provided with detailed usage instructions.

These baseline OS Build Plans have been tested for their specified platform and are known to work. You simply need to make a copy of a Build Plan for the appropriate platform and modify it for your specific environment (modifications include: location of the Media Server, required scripts, required reboots, etc.).

Once installed, Build Plans appear in the SA Client Library under:

```
/Opware/Tools/OS Provisioning/OS Build Plans/Windows  
/Opware/Tools/OS Provisioning/OS Build Plans/Linux
```

OS Build Plan Updates

On occasion, new baseline OS Build Plans may be released. You will be required to download these Build Plans from the HP Live Network and install them using the DCML (DET) tool. Installation instructions and detailed descriptions for each baseline OS Build Plan are included with the download. Installation steps will be similar to the following:

```
mkdir /tmp/osbp_import
unzip -d /tmp/osbp_import /var/tmp/OPSWosbp_content-40.X.Y.Z.zip
/opt/opsware/cbt/bin/cbt -cf /var/tmp/core.cfg -i /tmp/osbp_import
```

Sample OS Build Plan Usage Instructions

Instructions for using the sample OS Build Plans provided by SA are included in the SA online help. In the SA online help, click below to view the OS Build Plan instructions:

[View OS Build Plan instructions](#)

OS Build Plan Requirements

- During installation, SA by default installs two Automation Platform Extensions (APXs) that perform the actions specified in Build Plan: the Run OS Build Plan Web APX and Program APX . These APXs appear in the SA Client's APX Library and can be run manually or in scripts for certain tasks, but they should not be removed or modified.
- The OS Build Plan wizard requires that you install the Adobe Flash Player plug-in on all clients from which you will run the wizard.
- [Windows] In order for Windows OS Build Plan OS provisioning to copy/install folders and files from the OS media's \$OEM\$ folder to the mapped destination folder on the target server, you should use one of the following options:
 - Create a new OS Build Plan by copying a baseline Windows OS Build Plan and ensure that the unattend file section in the Configure Windows <version> Default Unattend.txt script has the following entry:

```
OemPreInstall=Yes
```

Add the Inject Required Unattend.txt setting OGFS script as a step in the OS Build Plan before the Mount Windows Share script.
 - Use one of the baseline Windows Default Install OS Build Plans and provide it with the name of an existing OS Installation Profile. In this case, the legacy behavior is emulated, so OemPreInstall=Yes is automatically added if missing.
- [Linux] SA provides Inject Kickstart personalization Settings for VMware ESXi 4.1 in the sample VMware ESXi 4.1 OS Build Plans. Direct network parametrization is supported, however, the out-of-box OS Build Plan SAMPLE: ESXi 4.1 Default Install with static IP provides the same result with less manual intervention than network parameterization. See the readme.html file included with the sample OS Build Plans for instructions about modifying the sample file for your specific installation.
 - If you have modified the network parameterization settings for a server and later run a baseline OS Build Plan against that server, your modified network parameterization settings are overwritten (this does not occur when using OS Sequences).

Permissions

You must have been granted permissions for OS Build Plans as described in the Permissions Reference appendix in the *SA Administration Guide*.

Custom Attribute Parameters and OS Build Plans

While provisioning servers using OS Build Plans you can use Custom Attributes in two contexts:

- Build Plan context: Custom Attributes that impact the overall provisioning with the specific Build Plan. See [Figure 21](#) for more details.
- Server context: Custom Attributes that impact the provisioning of a specific server. The attributes can be added at the following levels: **Server > Device Group > Customer > Realm > Facility > Operating System > Software Policies**.

Copying a Baseline OS Build Plan



You must always make a copy of the default baseline OS Build Plan content files. Never modify the original.

After you have downloaded and installed the baseline OS Build Plans, HP strongly recommends that you copy an appropriate plan and use that as the basis for your build plan.

To copy a plan:

- 1 Log in to the SA Client.
- 2 Navigate to **Library > OS Build Plans**.
- 3 In the OS Build Plan pane, right click and select **New** from the context Menu.
- 4 In the OS Build Plan Wizard, give your new OS Build Plan a name and optional description on the Properties page.
- 5 Select Build Plan Items and click the Copy Plan button and navigate to the folder in which you stored the baseline OS Build Plans.
- 6 Highlight the plan you want to copy and click Select. The task scripts contained in the baseline OS Build Plan are copied into the task list for your new plan.
- 7 Modify the script parameters for your environment. Note that some script have required parameters like @MediaServer@ for which you must supply the fully qualified path and filename for your media server.
- 8 Save your new OS Build Plan.

Alternatively:

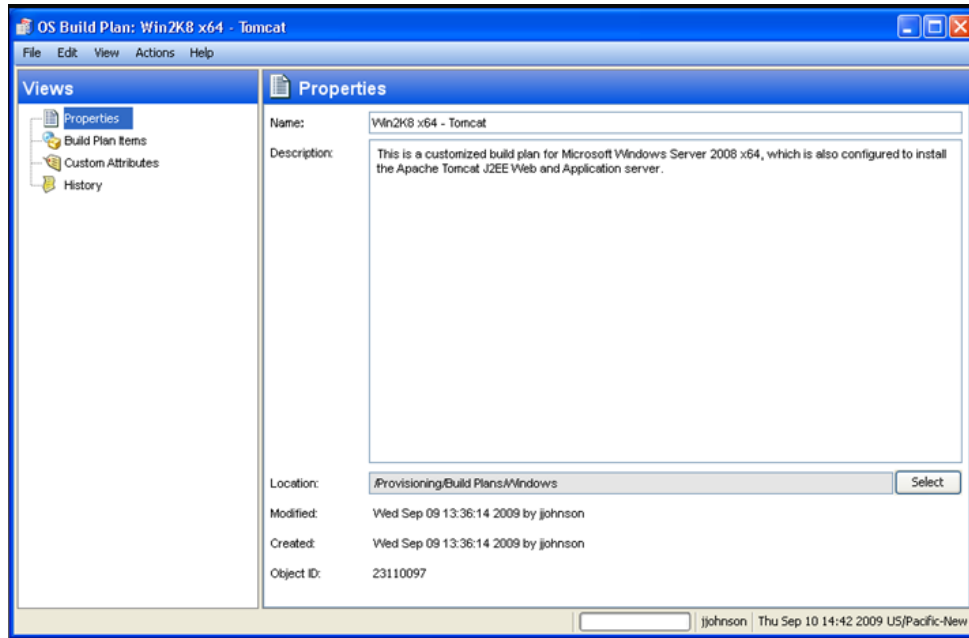
- 1 Log in to the SA Client.
- 2 In Navigation Plane open **Library > Tools > OS Provisioning > OS Build Plans > Windows**.
- 3 Highlight and copy the required build plan.
- 4 Paste and save the plan to a different folder and rename as required.
- 5 Open and modify the Build Plan for your environment.

Viewing/Modifying OS Build Plans

You view and modify OS Build Plans using the SA Client.

- 1 Log in to the SA Client.
- 2 From the Navigation pane, select the By Type tab, then select **Library > Tools > OS Provisioning > OS Build Plans > Windows** or the folder in which you saved a modified Build Plan. Right click in the OS Build Plans list pane and select **Open**.
- 3 SA displays the Build Plan Properties page.

Figure 19 Build Plan Properties Page

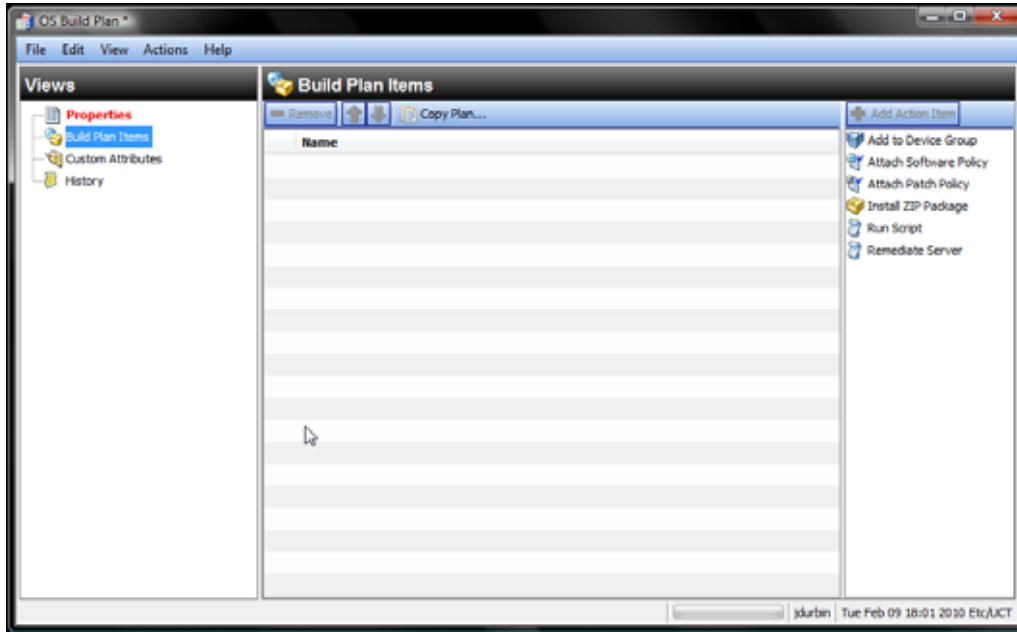


On this page, you see the OS Build Plan name and optional description. You can view other build plans using the Select button.

On this page you can also rename the current build plan or select a different plan. SA also displays information about the build plan itself including location, last modified date and the user id of the modifier, and the OS Build Plan's Object ID.

Select Build Plan Items in the View pane to display the Build Plan Items page in which you can see the tasks that have been assigned to the plan.

Figure 20 Build Plan Items Page



You use the Build Plan Items page to add and organize the tasks performed by your OS Build Plan.

The Action pane displays a list of actions you can add to your build plan. Double click on an action to add it to the plan. Use the green arrow keys to move actions up and down in the build plan order. To remove an action you have added, highlight the action and click Remove.

Actions available are:

- **Add to Device Group:** attaches the server to a Device Group.
- **Attach Software Policy:** attach a policy that specifies software to be installed. See “Software Management” in this guide for more information.
- **Attach Patch Policy:** attach a policy that specifies Windows patches to be applied to the server. See the *SA User Guide: Server Patching*. Adding a Solaris Patch policy to an OS Build Plan is not supported in this release.
- **Install Zip Package:** specify a zip package to install and the installation path.



You can specify the installation path for the zip package in the OS Build Plan. If you add a zip package to an OS Build Plan but *do not* specify the installation path in the plan, SA uses the installation path specified in the zip package.

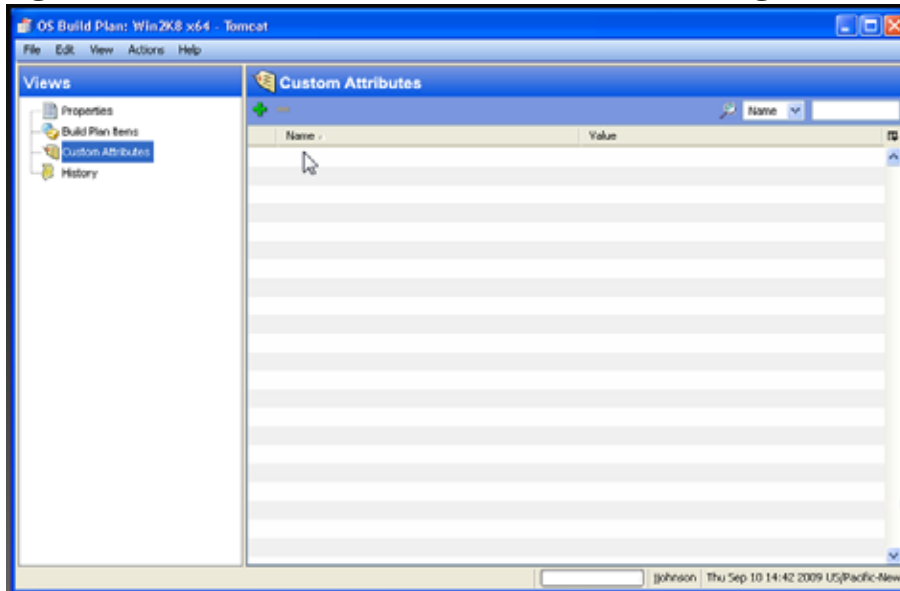
- **Run Script:** run Unix, OGFS scripts or Windows Visual Basic and BAT scripts.
- **Remediate Server:** specify the remediation's reboot options and Error Handling option.

For each action, you must use Select pane at the bottom to select the specific script, policy, device group, and on to assign for the action. For some actions, for example Remediate Server, you may be required to specify additional configuration information. An item that requires additional configuration information is indicated by a red exclamation point superimposed on its icon.

SA does not enforce the correct order for Build Plan tasks, you must ensure that you add all tasks in the correct order.

- 4 *This feature is not currently available.* Select Custom Attributes from the View pane to display the Custom Attributes parameters page.

Figure 21 Build Plan Custom Attribute Parameters Page



This page displays any custom attribute parameters that have been specified for the OS Build Plan.

While provisioning servers using OS Build Plans you can use Custom Attributes in two contexts:

- Build Plan context: Custom Attributes that impact the overall provisioning with the specific Build Plan. see [Figure 21](#) for more details
- Server context: Custom Attributes that impact the provisioning of a specific server. The attributes can be added at the following levels: **Server** > **Device Group** > **Customer** > **Realm** > **Facility** > **Operating System** > **Software Policies**.

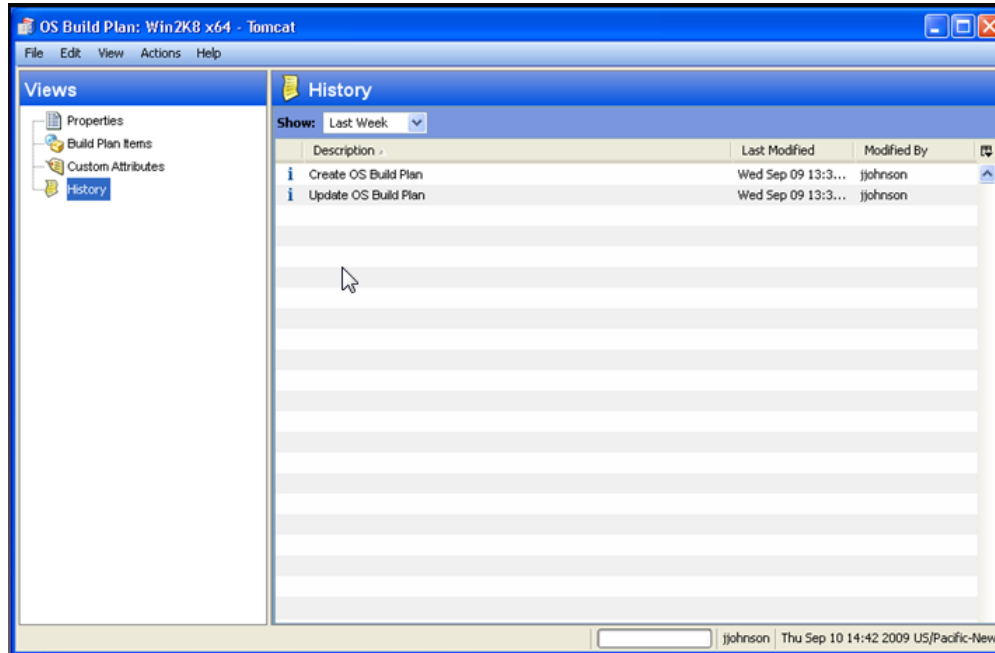
Click the green plus sign to add a new custom attribute parameter. Highlight an existing custom attribute and click the minus sign to remove it. You can also control the number of custom attributes displayed and search for specific attributes using the search box (wildcards supported).

When you select Add or select an existing build plan, you can specify or change the Name field and in the Value field, specify the values for the custom attribute. Clicking the ellipsis button causes the Custom Attribute editor to open. This simple editor allows you to more easily enter multiple custom attribute values.

The process of adding/modifying custom attributes is similar to that described in [Defining Custom Attributes](#) on page 83. See that section for more information about custom attributes.

- 5 Select History from the View pane to display the Build Plan History page, [Figure 22](#).

Figure 22 Build Plan History Page



This page displays a chronological history of changes made to the OS Build Plan.

Minimum Baseline OS Build Plan Modification

HP strongly recommends that you copy a baseline OS Build Plan and use it as the basis for your own build plan by modifying it for your environment. Do the same for any scripts you plan to modify for your own environment.

The following are the minimum modifications you must make to a baseline OS Build Plan in order to customize it for your environment.

- **Media Server Host:** the fully qualified path to the OS Provisioning Media Server host. The default is `MediaServer`. Specify from the OS Build Plan Items page, Install OS Media.
- **Product Key Custom Attribute:** using the Custom Attributes page shown in [Figure 21](#), you must specify the correct product key for your Windows product.
- **Path to Windows Setup:** provide the full path to the Windows installer executable program.
- **[WIM only] Path to the WIM Image:** provide the full path to the WIM image.
- Any requirements specific to your OS Provisioning environment such as scripts to be run, reboots, remediation, etc.

You can now run your OS Build Plan against unprovisioned servers. For more information, see [Using an OS Build Plan for OS Provisioning](#) on page 127.

Creating OS Sequences

An OS Sequence defines what to install on a server, such as operating system configuration information taken from an OS Installation Profile that you specify, software and patch policies, and the target servers on which to install the operating system.

▶ When you create an OS Sequence, it is saved into the Folder list in the Library. You must have permissions to the folder where you want to save the OS Sequence. For more information on how folder permissions work, see “User and Group Setup and Security” in the *SA Administration Guide*.

OS Sequence Contents

You can specify the following in an OS Sequence:

- **Properties:** Allows you to name the OS Sequence and choose a location to save it in a library folder. You must have permissions to write to the folder where you save the OS Sequence, otherwise you will be unable to save it in the selected location in the library.
- **Install OS:** Allows you to choose an OS Installation Profile. If the OS Installation Profile already has a customer associated with it, you will be unable to select a customer for the OS Sequence. If it does not have a customer associated with it, then you can select one here. Once you choose a customer, then all servers on which you install the operating system using this OS Sequence will be associated with that customer.

Attach Patch Policies is available for Windows and Solaris OS Sequences.

For more about information Patch Management, see the *SA User Guide: Server Patching*.

- **Attach Device Group:** Allows you to select a device group (group of servers) for a the server once the OS Sequence has been run. You can select any public static group to attach to the OS Sequence.

A group of servers can also have software and patch policies associated with it. If you enable remediation in the OS Sequence (in Remediate Policies), then all software and patches associated with the group of servers will also be installed on the server when you run the OS Sequence. If you disable remediation, then none of the software or patches in the policies attached to the group of servers will be installed on the server.

For information on device groups, see Server Management in the *User's Guide: Server Automation*.

- **Remediate Polices:** Allows you to choose to enable or disable remediation when the server is provisioned with the OS Sequence. The Default is **Disabled**.

When remediation is disabled, running an OS Sequence installs the operating system however no policies in the OS Sequence are remediated—that is, no software or patches in any of the policies attached to the OS Sequence are installed when the sequence is run.

If you enable remediation, then all software and patches in all policies attached to the server will be installed when the OS Sequence is run. This is also true for any policies attached to the group of servers selected for the OS Sequence. You can also set reboot and pre and post installation script options.

▶ In order to perform OS Provisioning with remediation, you must have at minimum read access to all server module policies.

Defining an OS Sequence

To create an OS Sequence, perform the following steps:

- 1 In the SA Client, from the Navigation pane, select Library and then select OS Sequences.
- 2 Choose an OS folder.
- 3 From the **Actions** menu, select **New...**
- 4 In the Views pane of the OS Sequence window, select Properties and enter a name for the OS Sequence.
- 5 Click **Change** in the Content pane to choose a location in the folder library to save the OS Sequence. You must have permissions to write to the folder where you save the OS Sequence.
- 6 From the Views pane, click **Tasks** then **Install OS** to choose an OS Installation Profile.
- 7 If the OS Installation Profile does not have a customer associated with it, then select a customer from the Assign Customer drop-down list. If the OS Installation Profile already has a customer associated with it, you will be unable to select a customer for the OS Sequence. All servers provisioned with this OS Installation Profile will be associated with the specified customer (if a customer has been assigned).
- 8 From the Views pane, select **Attach Software Policy**.
- 9 At the bottom of the Content pane, click **Add** and select a software policy to add to the OS Sequence.
- 10 From the Views pane, select **Attach Patch Policies**.
- 11 At the bottom of the Content pane, click **Add** and select a patch policy to add to the OS Sequence.
- 12 From the Views pane, select **Attach Device Group**.
- 13 At the bottom of the Content pane, click **Add**. Select a device group to place the server into, after the OS Sequence has been run. You can only select a public static group for this option.
- 14 From the Views pane, select **Remediate Policies**.
- 15 In the Content pane, choose to enable or disable remediation when the server is provisioned with the OS Sequence. If you select Disable Remediation, then when you run the OS Sequence, the operating system will be installed but no policies in the OS Sequence will be remediated — this means that no software in any of the policies attached to the OS Sequence will be installed when the sequence is run.
- 16 If you select Enable Remediation, then you will need to configure the Rebooting and Scripts parameters. For the rebooting options, you can select one of the following:
 - **Reboot servers as dictated by properties on each installed item:** Selecting this option will allow any reboot settings to run that might be set in any software or patch policies attached to the OS Sequence.
 - **Hold all server reboots until after all items are installed:** This option will override any pre-install reboot options that might be set in any software or patch policies attached to the OS Sequence. If any post-install reboots have been set, then they will execute after the operating system has been installed.
 - **Suppress all server reboots:** This option will override reboot options set in any software or patch policies attached to the OS Sequence.

- 17 Next, in the Scripts section, select either a Pre-Install/Post-Install Script. These tabs allow you to set a pre- or post-install script to be executed before the OS Sequence has been run and after the operating system has been installed. Click **Enable Script** to enable a the script parameters.
- 18 From the Select drop-down list, select either Saved Script or Ad Hoc Script. Each script type has its own settings:

Saved Script

- **Command:** Add any commands or arguments to be executed here.
- **Script Timeout:** Enter a numerical value for the number of minutes to pass until the script will timeout.
- **User:** Enter a user name and password, or choose to run the script as Local System. (If using Unix, choose root as the user.)
- **Error:** Select if you want the OS Sequence job to stop if the script returns an error.

Ad Hoc Script

- **Type:** Choose UNIX shell for Unix systems, or for Windows, select BAT or VBSCRIPT.
 - **Script:** Enter the text of the script. An Ad-Hoc script runs only for this operation and is not saved in SA. In the Script box, enter the contents of the script.
 - **Command:** If the script requires command-line flags, enter the flags here.
 - **Script Timeout:** Enter a numerical value for the number of minutes to pass until the script will timeout.
 - **User:** Enter a user name and password, or choose to run the script as Local System account. (If using Unix, choose root as the user.)
 - **Error:** Select if you want the OS Sequence job to stop if the script returns an error.
- 19 When you have finished making your selections, from the **File** menu, select **Save** to save the OS Sequence.

5 Performing OS Provisioning

This chapter describes the process of provisioning an operating system to a server. It includes basic, step-by-step tutorials for common OS Provisioning tasks.

SA OS Provisioning-supplied CD Boot Images

SA OS Provisioning provides several service operating system boot CD images (ISOs) that you can use to record to CD. These ISO images can also be configured in virtual machine CD-ROM drives or mounted using iLO Virtual Media or similar technology. Use the SA Client Export utility to download the required image(s) and burn boot CDs.

These files are located in the SA Software Library Folder:

```
/Opware/Tools/OS Provisioning/
```

and are named using the format:

Linux:

```
HPSA_linux_boot_cd.iso
```

Windows:

```
OPSWwinpe<arch>.iso
```

Sun Solaris OS Provisioning Quick Start

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 operating system to be installed in your environment. OS Build Plans are not supported for Solaris.

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

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

- 1 Copy the Sun Solaris operating system media to the Media Server by using the scripts included on the Sun Solaris installation CD-ROM or DVD. See [Import Media Tool Prerequisites](#) on page 34.
- 2 Create an MRL for the Solaris media by using the Import Media tool. See [Creating Media Resource Locators \(MRLs\)](#) on page 33.

- 3 Create a Solaris JumpStart profile with a text editor. See [Sun Solaris Installation Profile Requirements](#) on page 51.
- 4 Prepare an OS Installation Profile for the Solaris operating system using the SA Client. Specify the location of the Solaris operating system media (with the MRL) and upload the profile. For more information, see [Defining an OS Installation Profile — Linux/Unix](#) on page 59.
- 5 (Optional) Customize the default build process that OS Provisioning uses to install the version of Solaris on servers. See [Solaris Build Customization Scripts](#) on page 72 and [Requirements for Solaris Build Customization Scripts](#) on page 74.
- 6 (Optional) Define custom attributes in the OS Installation Profile. See [Defining Custom Attributes](#) on page 83 and [Custom Attributes for Sun Solaris](#) on page 84.
- 7 Create an OS Sequence using the SA Client. See [Creating OS Sequences](#) on page 100.
- 8 Install the operating system as described in [Using an OS Sequence for OS Provisioning](#) on page 129.

Linux or VMware ESX OS Provisioning Quick Start

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 operating system (and template) to have a single configuration associated with it.

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

To set up Linux or VMware ESX OS Provisioning, perform the following tasks:

- 1 Copy the Linux or VMware ESX operating system media to the Media Server. See [Import Media Tool Prerequisites](#) on page 34 and [Creating Media Resource Locators \(MRLs\)](#) on page 33.
- 2 Create the operating system-specific configuration file (Jumpstart, kickstart, etc.) with a text editor. See [Configuration Files](#) on page 51.
- 3 Prepare required OS Provisioning configuration files:
 - If you are using an OS Sequence, prepare an OS Installation Profile for the Linux or VMware ESX operating system using the SA Client. Specify the location of the Linux or VMware ESX operating system media (with the MRL) and upload the configuration file. See [Defining an OS Installation Profile — Linux/Unix](#) on page 59.
 - (Optional) Customize the default build process that OS Provisioning uses to install Linux on servers. See [Using Build Customization Scripts](#) on page 71.
 - (Optional) Define custom attributes in the OS Installation Profile. You can edit the custom attributes so that the profile passes specific information to the Linux 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. See [Defining Custom Attributes](#) on page 83.

- (Optional) Add new hardware support to a Linux or VMware ESX build image. OS Provisioning includes build images that install the target operating system on servers for Linux or VMware ESX. For more information, see [Adding Hardware Support to a Linux or VMware ESX Build Image](#) on page 28.
 - If you are using an OS Build Plan, prepare the OS Build Plan as described in [Creating OS Build Plans](#) on page 90.
- 4 Install the operating system as described in [Using an OS Build Plan for OS Provisioning](#) on page 127 or [Using an OS Sequence for OS Provisioning](#) on page 129.

Microsoft Windows OS Provisioning Quick Start

To prepare for Windows OS Provisioning, you must set up a Windows unattended installation or use an OS Build Plan.

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

- A licensed copy of the Windows operating system 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 provisioning follows the general process for OS Provisioning setup. To set up Windows provisioning, perform the following tasks:

- 1 Copy the Windows operating system media to the Media Server. See [Import Media Tool Prerequisites](#) on page 34.
- 2 If you are using an OS Sequence, create an MRL for the Windows media by using the Import Media tool. See [Creating Media Resource Locators \(MRLs\)](#) on page 33.
- 3 Create a Windows response file with a text editor. See [Microsoft Windows Installation Profile Requirements](#) on page 53.
- 4 Prepare the OS Provisioning configuration files.
 - If you are using an OS Sequences, prepare an OS Installation Profile for the Windows operating system using the SA Client. See [Defining an OS Installation Profile — Windows](#) on page 60. This section explains how to specify the location of the Windows operating system media (with the MRL) and upload the response file.
 - (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 63.
 - (Optional) Customize the default build process that OS Provisioning uses to install the version of Windows on servers.

- Add custom attributes to the OS 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 [Defining Custom Attributes](#) on page 83.
 - If you are using an OS Build Plan, prepare the OS Build Plan as described in [Creating OS Build Plans](#) on page 90.
- 5 Install the operating system as described in [Using an OS Build Plan for OS Provisioning](#) on page 127 or [Using an OS Sequence for OS Provisioning](#) on page 129.

Booting Servers Remotely

On *Intel-based servers*, you can remotely boot a new server over a network using PXE. For other servers that do not support network boot technology, SA supports bootable CDs.

The SA Boot CD contains a small operating system, network drivers, the software required to mount a network drive and the required SA communications infrastructure.

For *Solaris servers*, you can provision an operating system over the network if DHCP is available.



To boot servers over the network, the installation client must either be able to communicate with the SA DHCP server on the SA Core network or, for operating systems on which it is supported, you must supply static network configuration information at boot time. If the installation client is running on a different network than the SA core network, your environment must have a DHCP proxy (IP helper).



Before attempting to boot any server using DHCP to the SA server pool, ensure that the DHCP server's configuration file has the following line uncommented:

```
authoritative;
```

Booting from a CD

You can boot a remote server in three ways:

- 1 Copy the downloaded ISO file into the ESX/ESXi hypervisor's datastore and mount to the VM's virtual CD-ROM drive
- 2 Use iLO Virtual Media or a similar out-of-band management technology provided by your server's hardware vendor.
- 3 Record (burn) the operating system's ISO image to a physical CD and load the CD into a CD-ROM drive on the target server)

Network Booting a Linux or VMware ESXi Server

The following section explains how to network boot a server to prepare for Linux, VMware ESX/ESXi 4.1 (OS Build Plans only) provisioning with PXE." For information on how to boot a Windows server with WinPE, see [Network Booting a Windows Server](#) on page 112. For more information about hardware support, see [Hardware Preparation](#) on page 25.

To boot a Linux, VMware ESX/ESXi 4.1 server (OS Build Plans only) using PXE, perform the following tasks:

- 1 Prepare the server and connect it to the SA network, configure the server to boot using PXE.

See the hardware vendor's documentation for information about configuring a server to boot using PXE.

- 2 Power on the server and select the option to boot the server using PXE.

- 3 The following menu is displayed.

```
winpe32-ogfs - Windows Build Agent (WINPE 32 bit - OGFS based)
winpe64-ogfs - Windows Build Agent (WINPE 64 bit - OGFS based)
winpe32      - Windows Build Agent (WINPE 32 bit)
winpe64     - Windows Build Agent (WINPE 64 bit)
linux5      - Linux Build Agent (RHEL 5.6-based)
linux6      - Linux Build Agent (RHEL 6.0-based)
linux6-ogfs - Linux Build Agent (RHEL 6.0-based - OGFS-based)
solaris     - Solaris x86 Build Agent
localdisk   - Normal boot from localdisk (default after 10 seconds)
```

Intel-based Linux or VMware ESXi PXE Booting

- a Choose an SA PXE boot image for OS Sequence provisioning by entering the appropriate text at the boot prompt:

- linux5 (Red Hat Enterprise Linux 5 and other supported Linux versions)
- linux6 (Red Hat Enterprise Linux 6)

Intel-based Linux or VMware ESXi OGFS Agent Booting

- b For OS Build Plan-based (OGFS Agent) Linux or VMware ESXi 4.1 provisioning, enter:

```
linux6-ogfs (Red Hat Enterprise Linux 6)
```

- 4 press enter to begin the boot process.



If the operating system you are provisioning is Red Hat Enterprise Linux 3 IA64, you must add the custom attribute `kernel_arguments` with the value `console=ttyS1` to the OS Installation Profile.

- 5 After the booting process finishes successfully, a message appears on the console indicating that the server is ready for OS Provisioning and the server now appears in the SA Client Unprovisioned Servers list as available for operating system installation.
- 6 (*Optional*) Record the MAC address and/or the serial number of the server so that you can locate the server in the SA Client Unprovisioned Servers list.

- 7 Verify that the server appears in the SA Client Unprovisioned Servers list and that it is ready to hand off for operating system installation.

See [Verifying That a Server is Ready for Operating System Installation](#) on page 118 in this chapter for more information.



You must reboot any servers running Linux OGFS Agents when you upgrade the core or restart the NFS services.

Booting a Red Hat Enterprise Linux Server in a Non-DHCP Environment

If you plan to use OS Provisioning in an environment without a DHCP server, you must assign static IP information for the managed server and manually configure that server to resolve the SA Core.

There are several reasons you might need to manually specify the network information for a sever being provisioned:

- You don't use DHCP and must manually specify the static IP address and the Agent's IP and Port
- You must provision a server but DHCP is inactive.
- You must provision a server but DHCP is blocked by firewall rules.

CD boot images for Linux OS provisioning in non-DHCP environments can be exported by selecting **Library > By Folder > Opsware > Tools > OS Provisioning**.

The images are named using the following format:

```
HPSA_linux_boot_cd.iso
```

This section provides details for provisioning in a non-DHCP environment.

When you boot an unmanaged server in a non-DHCP environment, you will see a boot screen similar to that shown in [Figure 23](#):

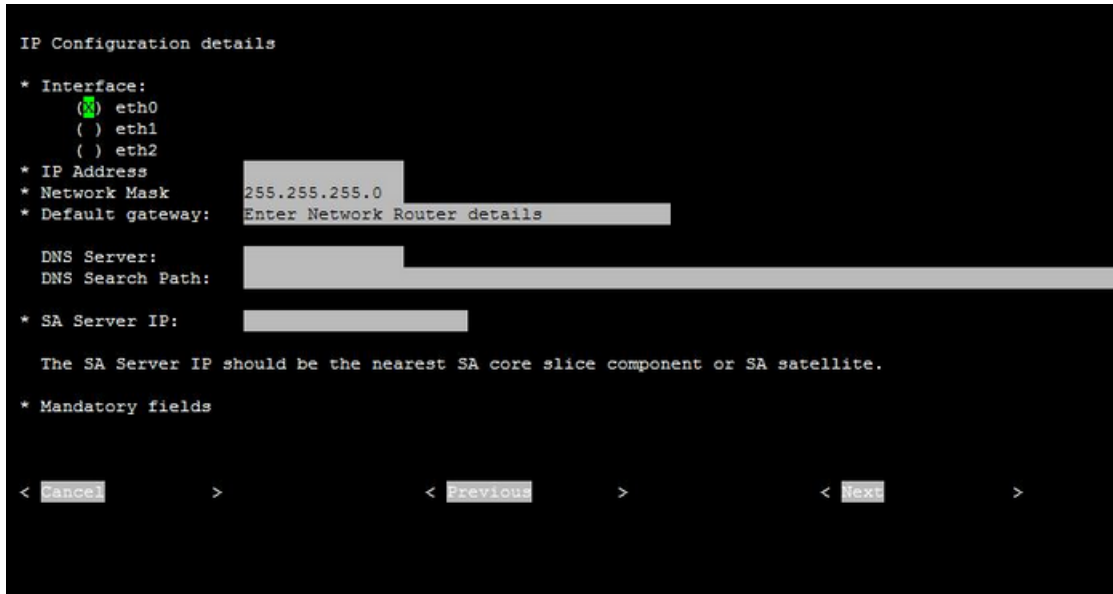
Figure 23 Red Hat Linux Boot Screen



After you select the boot method, you see a dialog that allows you to choose whether you want to boot your machine using DHCP or enter the static network configuration.

If you choose DHCP, SA uses your DHCP server for configuration. If you choose static, you will see a Network Configuration dialogue that allows you to enter a static IP address for the server, the subnet mask, The host gateway IP address, and the IP address and default port for the SA Agent Gateway, [Figure 24](#):

Figure 24 Red Hat Linux Network Configuration Dialog



You can manually configure the following fields:

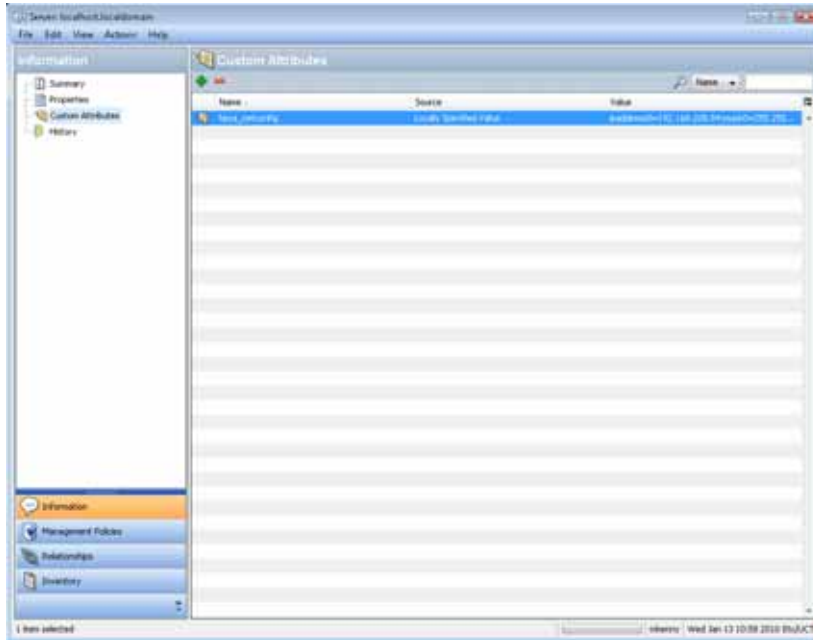
- Interface: the NIC to be used
- IP Address: static IP address for the server being provisioned
- Netmask: netmask for the server being provisioned
- Default gateway: Gateway IP address the server being provisioned should use (network level IP router)
- DNS Server: the IP address the server being provisioned should use
- DNS Search Path: the fully qualified DNS suffix the server being provisioned should use
- SA Server IP: IP address of the SA Core host

After the information in these fields is entered and applied, the server is able to register with the SA Core. You can now start the normal OS Provisioning process.

DHCP Custom Attribute

Servers that have been registered with the SA Core using a static IP specification will display the `hpsa_netconfig` custom attribute in the server record, as shown in [Figure 25](#):

Figure 25 hpsa_netconfig Custom Attribute in Server Record



Booting a Red Hat Enterprise Linux Itanium 64-bit Server in a Non-DHCP Environment

If you plan to use OS Provisioning in an environment without a DHCP server, you must assign static IP information for the managed server and manually configure that server to resolve the SA Core.

There are several reasons you might need to manually specify the network information for a sever being provisioned:

- You don't use DHCP and must manually specify the static IP address the Agent's IP and Port
- You must provision a server but DHCP is inactive.
- You must provision a server but DHCP is blocked by firewall rules.

You can export the Linux Itanium image by logging in to the SA Client and selecting **Library > By Folder > Opware > Tools > OS Provisioning**.

The images are named using the following format:

```
HPSA_linux_boot_cd_IA64.iso
```

The following section provides details for provisioning in a non-DHCP environment.

When you boot an unmanaged server in a non-DHCP environment, you will see a boot screen similar to that shown in [Figure 26](#):

Figure 26 Red Hat Linux Itanium 64-bit Boot Screen

```
Choose an Opware boot image by entering
the appropriate label at the 'EliLO boot:' prompt.

linux      - Linux 3 Build Agent
linux-txt  - Linux 3 Build Agent for serial consoles.
linux4     - Linux 4 Build Agent (default after 10 seconds)
linux4-txt - Linux 4 Build Agent for serial consoles.
linux5     - Linux 5 Build Agent
linux5-txt - Linux 5 Build Agent for serial consoles.

ELILO boot: 1
```

After you select the boot method, you will see a Network Configuration dialogue that allows you to enter a static IP address for the server, the subnet mask, The host gateway IP address, and the IP address and default port for the SA Agent Gateway, [Figure 27](#):

Figure 27 Red Hat Linux Itanium 64-bit Network Configuration Dialog



If the operating system you are provisioning is Red Hat Enterprise Linux 3 IA64, you must add the custom attribute `kernel_arguments` with the value `console=ttyS1` to the OS Installation Profile.

You can manually configure the following fields:

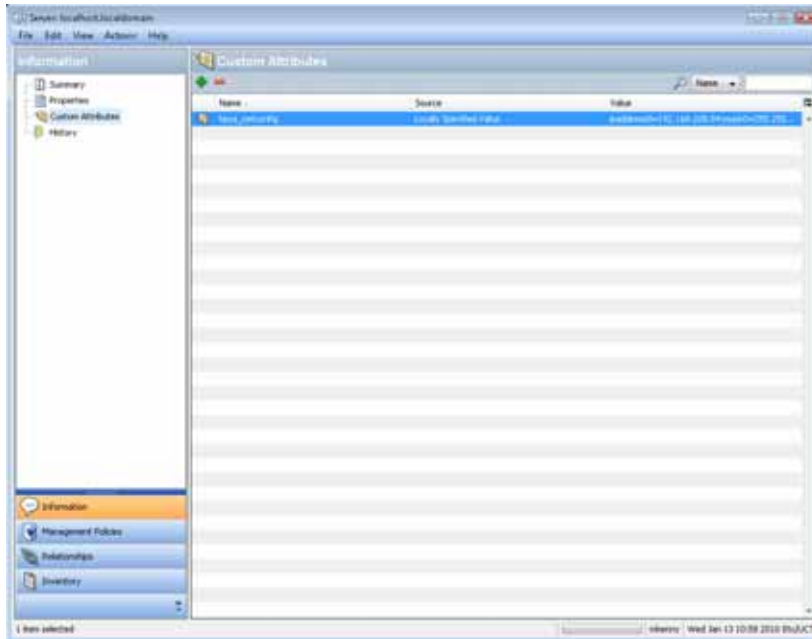
- Interface: the NIC to be used
- IP Address: static IP address for the server being provisioned
- Netmask: netmask for the server being provisioned
- Gateway: Gateway IP address the server being provisioned should use (network level IP)
- DNS Suffix: the fully qualified DNS suffix the server being provisioned should use
- Agent Gateway IP: the default SA Agent Gateway hostname or IP address
- Agent Gateway Port: the port used for the SA Agent Gateway

After the information in these fields is entered and applied, the server is able to register with the SA Core. You can now start the normal OS Provisioning process.

DHCP Custom Attribute

After provisioning, servers that have been registered with the SA Core using a static IP specification will display the `hpsa_netconfig` custom attribute in the server record, as shown in [Figure 28](#):

Figure 28 hpsa_netconfig Custom Attribute in Server Record



Network Booting a Windows Server

OS Provisioning supports booting a server with no operating system by PXE booting into a WinPE preinstallation environment or by booting into an OGFS-based environment. You can choose between either a WinPE x86 32-bit environment, a WinPE x86 64-bit environment (supports OS Sequences), or WinPE x86 32-bit and 64-bit using an OGFS-specialized Server Agent that boots from an SA-supplied OGFS Agent boot image (supports OS Build Plans).

In the WinPE x86 32-bit and WinPE x86 64-bit environments, an OS Provisioning Build Agent is loaded onto the server to be provisioned. In the WinPE-OGFS environment, the agent is a full SA Server Agent that is booted when an OGFS-enabled image is selected and is specialized for OGFS functionality to run OS Build Plans.

For more information about how the various OS Provisioning components like Build Agents and OGFS Server Agents function, see [SA OS Provisioning Components](#) on page 15.

WinPE also allows WIM-based image installation, as an alternative to unattended Windows installations.



You can PXE boot a Windows server using DHCP if that server resides on the same VLAN as an SA Boot Server (from an SA Core or Satellite). If your Windows server is not on a VLAN with PXE available, you can use a DHCP Helper IP relay configuration in your network switch. If neither configuration for network booting is available, you can boot using a CD-ROM

Windows PXE Booting

To boot a bare metal server with PXE into a WinPE preinstallation environment, perform the following steps:

- 1 Mount the new server in a rack and connect it to the SA build network.
- 2 Configure the server to boot using PXE.

See the hardware vendor's documentation on how to prepare a server to boot using PXE.

- 3 Power on the server and select the option to boot the server with PXE.

The SA menu appears and prompts you to select the type of OS Build Agent to load on the server.

All of the SA boot image options are displayed:

```
winpe32-ogfs - Windows Build Agent (WINPE 32 bit - OGFS based)
winpe64-ogfs - Windows Build Agent (WINPE 64 bit - OGFS based)
winpe32      - Windows Build Agent (WINPE 32 bit)
winpe64     - Windows Build Agent (WINPE 64 bit)
linux5      - Linux Build Agent (RHEL 5.6-based)
linux6      - Linux Build Agent (RHEL 6.0-based)
linux6-ogfs - Linux Build Agent (RHEL 6.0-based - OGFS-based)
solaris     - Solaris x86 Build Agent
localdisk   - Normal boot from localdisk (default after 10 seconds)
```

- 4 At the boot prompt enter:

```
winpe32
```

or

```
winpe64
```



If you do not select an option within 10 seconds, the server defaults to booting from the local disk. If you need more than 10 seconds to make your decision you can type any partial or invalid entry but do not press ENTER at the command line.

- 5 Press enter to begin the boot process.
- 6 A new menu displays the option to boot a WinPE x86 32 bit environment or a Windows x64 64 bit environment. Make a selection by using the arrow keys to highlight your choice, and then press ENTER.

The server will now be booted with the WinPE preinstallation environment. This may take a few minutes to complete, depending upon the speed of the network and the machine.

Once booting has finished, a new window will appear indicating that the server has had an SA Build Agent loaded and registered with the SA core.

- 7 *(Optional)* Record the MAC address and/or serial number of the server so that you can locate the server in the Server Pool list in the SA Web Client or in the Unprovisioned Servers list in the SA Client.
- 8 Verify that the newly racked server shows up in the SA Client Unprovisioned Servers, or SA Web Client Server Pool, and is ready for OS installation. See [Verifying That a Server is Ready for Operating System Installation](#) on page 118 in this chapter for more information.

Windows OGFS Agent Booting

- 1 Mount the new server in a rack and connect it to the SA build network.
- 2 Power on and boot the server with the Windows OGFS-based image.

The SA menu appears and prompts you to select the type of OGFS Agent to load on the server.

All of the SA boot image options are displayed:

```
winpe32-ogfs - Windows Build Agent (WINPE 32 bit - OGFS based)
winpe64-ogfs - Windows Build Agent (WINPE 64 bit - OGFS based)
winpe32      - Windows Build Agent (WINPE 32 bit)
winpe64      - Windows Build Agent (WINPE 64 bit)
linux5       - Linux Build Agent (RHEL 5.6-based)
linux6       - Linux Build Agent (RHEL 6.0-based)
linux6-ogfs  - Linux Build Agent (RHEL 6.0-based - OGFS-based)
solaris      - Solaris x86 Build Agent
localdisk    - Normal boot from localdisk (default after 10 seconds)
```

- 3 At the boot prompt enter:

```
winpe32-ogfs
```

or

```
winpe64-ogfs
```



If you do not select an option within 10 seconds, the server defaults to booting from the local disk. If you need more than 10 seconds to make your decision you can type any partial or invalid entry but do not press ENTER at the command line.

- 4 Press Enter to begin the boot process.
- 5 The server will now be booted with the OGFS-based preinstallation environment. This may take a few minutes to complete, depending upon the speed of the network and the machine.
- 6 Once booting has finished, a new window will appear indicating that the server has had an SA BOGFS Agent loaded and registered with the SA core.
- 7 Verify that the newly racked server shows up in the SA Client Unprovisioned Servers, or SA Web Client Server Pool, and is ready for OS installation. See [Verifying That a Server is Ready for Operating System Installation](#) on page 118 in this chapter for more information.

Booting a Windows Server in a Non-DHCP Environment

If you plan to use OS Provisioning in an environment without a DHCP server, you must assign static IP information for the managed server and manually configure that server to resolve the SA Core.

There are several reasons you might need to manually specify the network information for a server being provisioned:

- You don't use DHCP and must manually specify the static IP address and the Build Manager's IP and Port
- You must provision a server but DHCP is inactive.
- You must provision a server but DHCP is blocked by firewall rules.

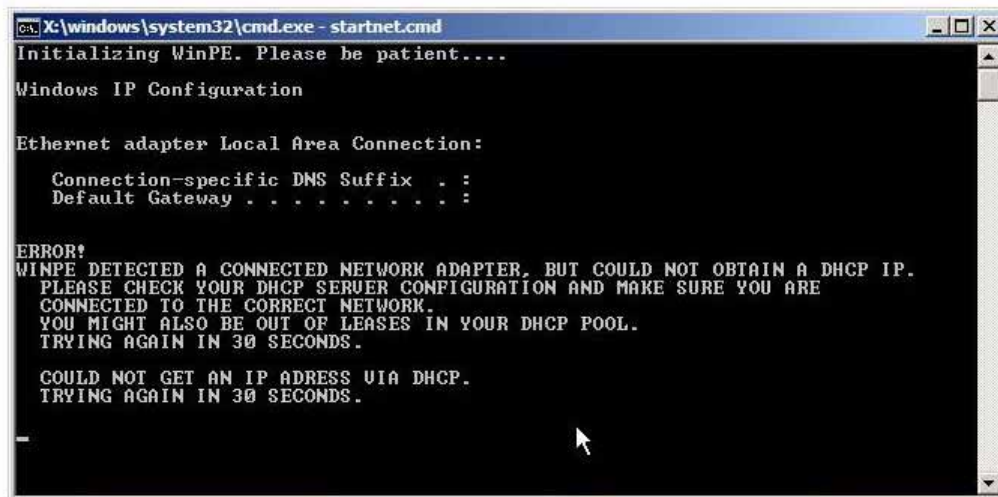
When provisioning a server using WinPE, by default, WinPE looks for a DHCP server. If a DHCP server is not found, you are prompted to enter the IP address, Subnet mask, Gateway and Name server of the host, and the Port and Hostname/IP of the SA Core.

This section provides details for provisioning in a non-DHCP environment.

Booting an Unmanaged Windows Server in a Non-DHCP Environment

When you boot an unmanaged server into a non-DHCP environment, by default WinPE looks for an available DHCP server. If WinPE does not find a DHCP server, you see a display similar to [Figure 29](#).

Figure 29 WinPE Console Display when DHCP Server Not Found



```
cs: X:\windows\system32\cmd.exe - startnet.cmd
Initializing WinPE. Please be patient...
Windows IP Configuration

Ethernet adapter Local Area Connection:

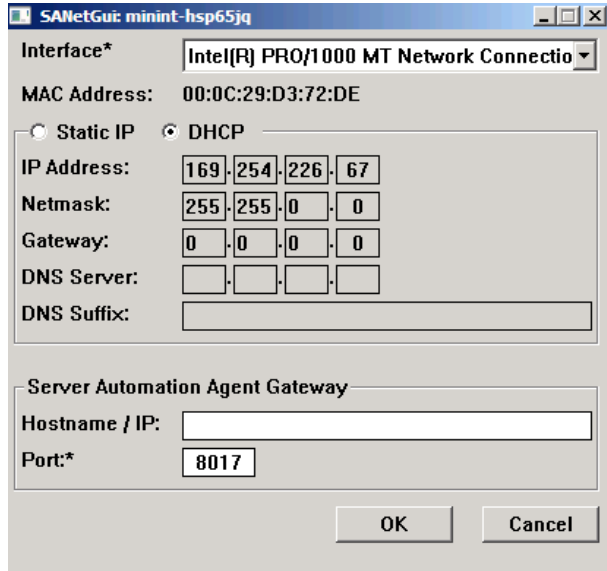
    Connection-specific DNS Suffix  . :
    Default Gateway . . . . . :

ERROR!
WINPE DETECTED A CONNECTED NETWORK ADAPTER, BUT COULD NOT OBTAIN A DHCP IP.
PLEASE CHECK YOUR DHCP SERVER CONFIGURATION AND MAKE SURE YOU ARE
CONNECTED TO THE CORRECT NETWORK.
YOU MIGHT ALSO BE OUT OF LEASES IN YOUR DHCP POOL.
TRYING AGAIN IN 30 SECONDS.

COULD NOT GET AN IP ADDRESS VIA DHCP.
TRYING AGAIN IN 30 SECONDS.
```

At this point, you will see a Network Configuration dialogue that allows you to enter the SA Agent Gateway IP or enter a static IP address for the server, the subnet mask, The host gateway IP address, and the IP address and default port for the Build Manager. See [Figure 30](#):

Figure 30 WinPE Network Configuration



Select the correct Interface and specify the Static IP.

You can manually configure the following fields:

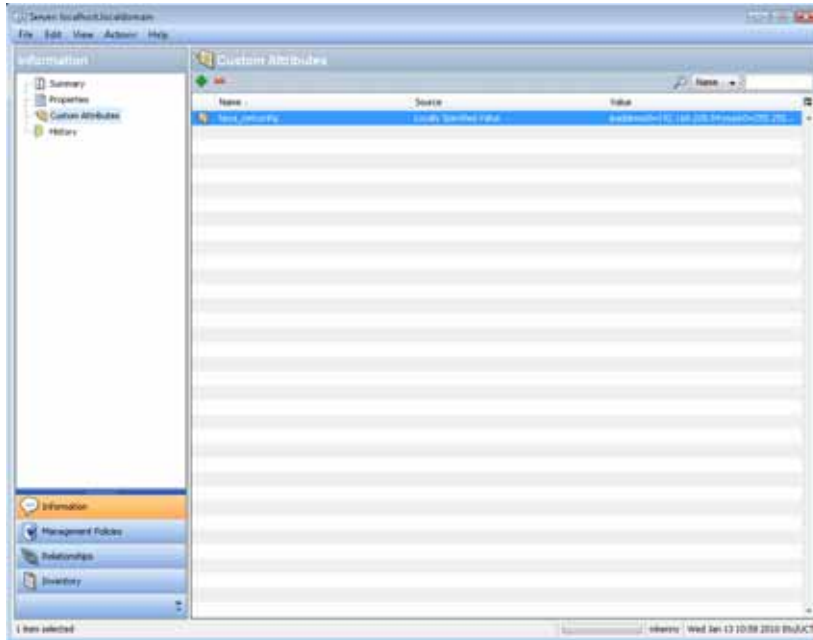
- IP Address: static IP address for the server being provisioned
- Subnet: Subnet mask for the server being provisioned
- Gateway: Gateway IP address the server being provisioned should use (network level IP router)
- DNS Server: the IP address the server being provisioned should use
- DNS Suffix: the fully qualified DNS suffix the server being provisioned should use
- Agent Gateway: SA Agent Gateway hostname or IP address
- Port: The port used for the Build Manager

After the information in these fields is entered and applied, the server being provisioned will be able to register with the SA Core.

DHCP Custom Attribute

Servers that have been registered with the SA Core using a static IP specification will display the `hpsa_netconfig` custom attribute in the server record, as shown in [Figure 31](#):

Figure 31 hpsa_netconfig Custom Attribute in Server Record



Network Booting Sun Solaris SPARC-based Server

When SA is installed, OS Provisioning is configured so that the Boot Server listens for broadcast requests from new servers and responds using DHCP.

Perform the following steps to boot a Solaris SPARC server over the network:

- 1 Mount the new Solaris SPARC server in a rack and connect it to the network.

The installation client on this network must be able to communicate with the SA DHCP server on the SA core network. If the installation client is running on a different network than the SA core network, your environment must have a DHCP proxy (IP helper).

- 2 Enter one of the following commands at the prompt:

```
ok boot net:dhcp - install
```

or

```
ok boot net:dhcp - install <interface_setting>  
<buildmgr=hostname|IP_address>
```

where *<interface_setting>* is one of the following options:

```
autoneg, 100fdx, 100hdx, 10fdx, 10hdx
```

You can include an interface setting with the boot command to set the network interface to a specific speed and duplex during OS Provisioning. Specifying this boot argument allows you to override the default interface setting that was specified when SA was installed in the local facility.

You can use a variety of methods including Solaris SPARC build customization scripts or specifying the values in a Solaris Package or RPM in the operating system media to set the network interface with a specific speed and duplex.

See [Creating Build Customization Scripts](#) on page 71 for more information.

Loading OS Build Agents

You can load an OS Build Agent on a server by booting the server with PXE for Intel-Based machine, Elilo for Itanium-Based machine or by using the network (Solaris). After a successful installation, the server appears in the Server Pool list.

You should verify that the newly racked server shows up in the SA Client **Unprovisioned Servers** list, or SA Web Client **Server Pool**, and is ready to hand off for operating system installation.

The SA Client's **Unprovisioned Servers** list and the SA Web Client **Server Pool** list display the servers that have registered their existence with SA but do not yet have an operating system installed.

You can start the operating system installation process in either one of two ways:

- From the SA Client's **Unprovisioned Servers** list, right click on the server in the content pane, and choose Run OS Sequence. See [Installing \(Provisioning\) an Operating System](#) on page 125 for details.
From the SA Web Client **Server Pool**, select the server and click **Install OS**. This option is only available for SA version 6.1 cores and later.



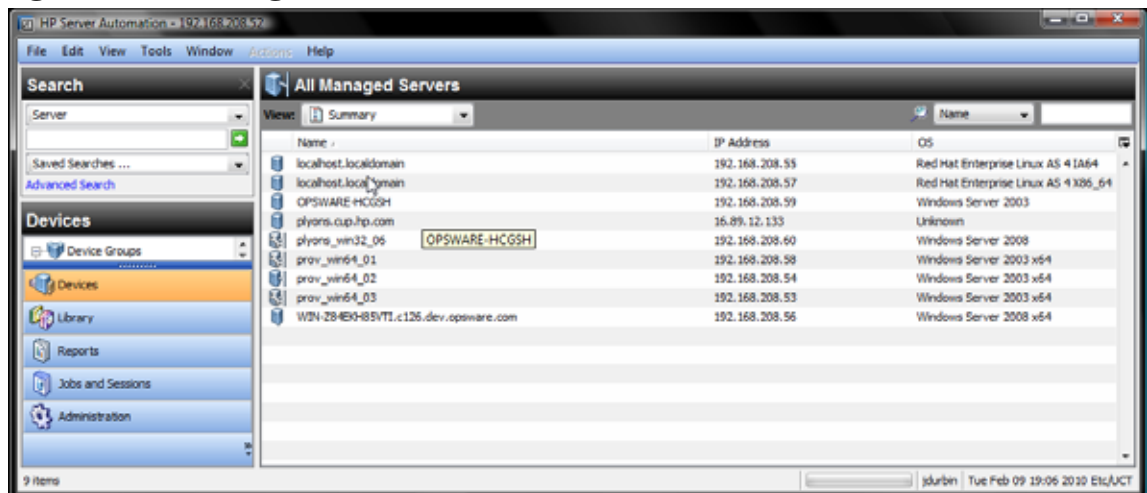
You must reboot any servers running OS Provisioning OGFS Agents when you upgrade the core or restart the NFS services.

Verifying That a Server is Ready for Operating System Installation

Perform the following steps to verify that a server is ready for operating system installation:

- 1 Log into the SA Client.
- 2 Select Devices in the Navigation pane, expand Servers and select All Managed Servers. The Managed Server page appears, as shown in [Figure 32](#). Servers in MANAGED/OK state are ready for OS system provisioning.

Figure 32 All Managed Server List in the SA Client



- New bare metal servers booted with an SA Build Agent (OS Sequence) are listed in the Unprovisioned Server pool(Unprovisioned mode)

- New bare metal servers booted with an SA OGFS Agent (OS Build Plan) are listed in the Unprovisioned Server pool (Maintenance mode)
- Unprovisioned servers booted with an SA OGFS Agent (OS Build Plan) are displayed in the Unprovisioned Server pool (Maintenance mode)
- Managed servers booted with an SA OGFS agent (OS Build Plan) stay in the Managed Server pool (Maintenance mode)

The Manage Boot Clients (MBC) Option

The Manage Boot Clients (MBC) option provides several services. You can:

- Remotely boot a server. You do not need console access to the server.
- Pre-create server records.
- Create custom attributes that set server configuration during OS Provisioning.
- Reconfigure services like DHCP when new servers are provisioned.
- Initiate OS Provisioning from a portal or an automated script where, typically, the user will not be available for interactive responses.

For example, you can change the default PXE image that a server uses to boot, change whether a server is assigned a DHCP lease, or specify the DHCP IP that is assigned to the server. You can also change a server's behavior when it enters the server pool, such as automatically invoking an OS Sequence when it enters the pool.

If the server is an HP ProLiant server with iLO2 enabled, and you know its iLO information, MBC can also remotely power on the server.

Any user, such as a system administrator who performs OS Provisioning and who is responsible for the base operating system, system utilities, patching, and the hand off of servers to internal business units, will find MBC quite useful.

You can access MBC functionality:

- From the SA Client
- From the Global File System command line
- From a script
- From a browser/portal form

Requirements

- The OS Provisioning infrastructure relies on SA Boot Server services for the MBC extensions.
- The OS Provisioning boot images must be served by the TFTP server that is shipped with SA.
- In order to take advantage of the DHCP reconfiguration feature, you must use the SA DHCP server.
- On a newly installed SA Core, a new user prior to running the MBC Web APX must first be granted Launch Global Shell permissions and must log in to the OGSF at least once in order to initialize the user environment (so that MBC can write temporary files to the user's home directories during use).

Required Permissions

In order to execute MBC, a user must have the *Allow Execute OS Sequence, Managed Server and Groups, Manage Customers, Server Pool, Read & Write permission to customer Not Assigned* and *Allow Configuration of Network Booting* permissions, write access to all pre-existing servers they will act on, and permissions to run the MBC APXs (thus, they need execute access on the `/Opware/Tools/OS Provisioning/Manage Boot Clients` folder).

Installation

The HP BSA Installer creates the MBC APXs during the SA Core installation. The installer creates a folder containing the MBC APXs in the SA Web Client Library, and adds an MBC Configuration Software Policy as part of the data baseline.

The following four APXs are installed for MBC:

- Program APX
- Web APX
- Integration Hook APX
- DHCP Cleanup Web APX

Using the Manage Boot Clients (MBC) Option

When MBC runs, it creates new server record(s) in the SA database in the Planned lifecycle. These records are displayed with a *blueprint* icon and can optionally have custom attributes assigned to them. Some of these custom attributes change how SA handles a server or configuration of an operating system installation (for example, you can set the **ComputerName** for a Windows unattended installation).

Executing MBC will typically change the default PXE menu choice when the server PXE boots, so that the user does not need to choose a PXE image on console of the server that's booting up. MBC also allows users to associate an OS Sequence with the server record so that, when the server registers as an unprovisioned server with HP SA, a provisioning job is kicked off automatically. Running an MBC APX from the SA Client

The MBC Web Interface

You can launch MBC Web APXs in three ways:

From the SA Client

- Select **Library ► Extensions ► Web ► Manage Boot Clients Web APX**.
- or, from the Unprovisioned Servers list, right click in the server list pane (not directly on a server) and select **Manage Boot Clients**.

From a Browser

You can also use a browser and navigate to:

`https://occ.example.com/webapp/osprov.manage_boot_clients_web/`

where `occ.example.com` is the local hostname or IP address for your SA Core.

The browser interface allows you to choose whether to use a form to input data for a singular host, or whether to input a CSV to set up multiple server records. After clicking the **Submit** button, it is grayed out to prevent double-submissions and a combined Progress/Results page is displayed.

The MBC Form-Based Method (Web-based)

The Web form-based interface provides a set of four pages that guide you through setting up an MBC job. You provide the information necessary to boot and provision a server on the first three pages/forms. The final page displays the progress/results of the job. You can act only on a single server when using the form-based method. For multiple server setup, you must use the CSV method.

Using the CSV Method from the Web Interface

The CSV input method can be accessed by clicking the **Multiple Client Form...** button on the first page of the MBC Web UI. The CSV input form allows acting on multiple server records at once, where each line in the CSV represents a server record.

The MBC APX Command-Line Interface

MBC also provides a Program APX, which is available to users as an executable in the Global Shell (OGSH). This can be useful for programmatic access to MBC while integrating with other systems.

Usage:

Users who have the appropriate permissions can run MBC from OGSH with this command:

```
/opsw/apx/bin/osprov/manage_boot_clients_script
```

Running MBC from the command line with no arguments will provide a usage statement.

This is an example command line entry that executes MBC and uses an existing CSV file:

```
/opsw/apx/bin/osprov/manage_boot_clients_script -m import  
<full path to CSV file with boot clients>
```

Special Attributes for the CLI and CSV Input Form

There are several special attributes which are not stored as custom attributes (except `sequence_id`) when entered, but instead are dealt with in distinct ways. [Table 12](#) lists these special attributes and how they are dealt with.

Table 12 MBC Special Attributes for the CLI and CSV Input Form

Parameter	Description
<code>pxe_image</code>	Specifies a PXE configuration files for the server. The value should be set to one of the options seen in the default PXE menu (such as <code>winpe32</code> , <code>winpe64</code> , or <code>linux5</code>). This copies the configuration file <code>/opt/opsware/boot/tftpbboot/pxelinux.cfg</code> to the MAC address file.
<code>sequence_id</code>	If specified, will invoke an OS Sequence installation (as <code>detuser</code>) as soon as the server is added to the Server Pool. Note: <code>sequence_id</code> actually is stored as a custom attribute on the server. This custom attribute is removed from the server record before the first reboot of the server.
<code>customer</code>	Sets the customer association for the server.
<code>use</code>	Sets the use field for the server. The value specified should be all caps (for example, <code>PRODUCTION</code>)
<code>stage</code>	Sets the stage field for the server. The value specified should be all caps (for example, <code>IN DEPLOYMENT</code>)
<code>facility</code>	Sets the facility association for the server. This is necessary when you run an MBC APX from a facility other than the one that the target server is associated with (necessary when you have a satellite that defines its own facility).
<code>ilo.*</code>	See “iLO Integration”.
<code>dhcpcleanup</code>	Retrieves the DHCP configuration or deletes a DHCP entry by MAC address. Options are: <ul style="list-style-type: none">• - help (-h): Displays online help• - action (-a): Options are:<ul style="list-style-type: none">— get: Retrieves the DHCP configuration— delete: Deletes a DHCP server(s) from the configuration. You must also specify either the:<ul style="list-style-type: none">— The MAC addresses or— <code>--facility (-f)=FACILITYNAME</code>: specify which facility's DHCP servers to operate on.— <code>--macs (-m)=MACS</code>: a comma-separated list of MAC addresses to remove from the DHCP configuration.— <code>--outputdir (-o)=OUTPUTDIR</code>: when specified, MBC saves progress and results information in the specified directory.

Additional non-MBC-specific custom attributes are available for the installation of Windows, Solaris, and Linux operating systems such as `hostname`, `ComputerName`, etc.

CSV Input Files

MBC's ability to accept CSV input files allows you to move servers into the Managed Server Pool and provision them with an operating system without the use of a console and an interactive session.

For example:

```
00:0c:29:e1:28:2e,hostname=testvm1,pxe_image=linux5,
sequence_id=2110061
00:0c:29:f9:12:f3,hostname=testvm2,pxe_image=winpe32
00:0c:29:0d:ab:b4,pxe_image=solaris,sequence_id=2110061
```

These CSV entries would cause MBC to create three Planned Server records and set them up to boot to the `linux5`, `winpe32`, and `solaris` PXE images, respectively. The servers processed by the first and third CSV entries will also have an OS Sequence applied when they register with SA. The first two entries would have specific display names shown in SA (`hostname=`), while the third would have an auto-generated hostname that be similar to `dhcp-client-00:0c:29:0d:ab:b4`. For more information on these attributes and their function, see the Special Attributes in [Table 12](#).

Example CSV Entries

```
00:13:E8:9A:93:BA,pxe_image=winpe32,dhcp.ip=10.2.3.11,
dhcp.hostname=m0011,customer=WealthManagement,
sequence_id=2030001,dns_server=10.6.4.2,
kernel_arguments=noacpi,root_password=wealth

00:13:E8:9A:93:BC,pxe_image=winpe32,dhcp.ip=10.2.3.12,dhcp.hostname=m0012,
customer=WealthManagement,sequence_id=2030001,
dns_server=10.6.4.2,kernel_arguments=noacpi,
root_password=wealth

00-13-E8-9A-93-99,pxe_image=linux

00:13:E8:9A:93:AA,pxe_image=windows,custattr1=val1,
custattr2=val2

00:13:E8:9A:93:BB,pxe_image=windows,customer=Opsware

00:0c:29:23:a1:7f,pxe_image=linux,sequence_id=310005,
testca=testval

00:0c:29:af:46:6b,pxe_image=linux,sequence_id=310005,
testca=testval

00:0c:29:be:96:6e,pxe_image=winpe32,sequence_id=320005

00-13-21-DD-DD-24,pxe_image=linux,sequence_id=310001,
dhcp.hostname=danube,ilo.hostname=10.128.32.102,
ilo.username=Administrator,ilo.password=adminpass,
ilo.reboot_if_on=1
...
```

The first item on each line of CSV must be a MAC address followed by a list of arbitrary, comma-separated name/value pairs, where the names and values are separated by equal signs. Each of these name/value pairs is stored as a custom attribute on the server record which allows the user to set up many custom attributes simultaneously.

Special Attributes for DHCP Reconfiguration

MBC has the ability to add host definitions to SA DHCP configuration files. This is useful in environments where SA DHCP is used, but configured to deny unknown clients (that is, it will only provide DHCP leases to *approved* MAC addresses). When you specify a DHCP hostname's MAC address on the **General** Form, MBC adds this MAC address to DHCP configuration. You can also specify DHCP IP address if required.

Table 13 lists the DHCP reconfiguration special attributes you can use in the CSV:

Table 13 DHCP Reconfiguration Special Attributes

Attribute	Description
dhcp.hostname	Specifies the MAC address for hostname(s) that are authorized for DHCP leases.
dhcp.ip	Specifies the IP address(es) of hosts that are authorized for DHCP leases.

iLO Integration

MBC includes integration with the HP Integrated Lights-Out 2 (iLO2) Standard. This increases the level of control that SA has over servers, down to the level where the users no longer have to even power on the servers. When the user provides an iLO IP and credentials, MBC will connect to the iLO API and automatically power on the server. ILO also provides more thorough hardware discovery.

Table 14 show the special attributes used for ILO Integration:

Table 14 ILO Special Attributes

Special Attribute	Description
ilo.hostname	Hostname or IP address for the iLO. This must be accessible from the hub/OGFS server. This value is stored as a custom attribute by MBC.
ilo.username	Username to use to authenticate to the iLO. This value is stored as a custom attribute by MBC.
ilo.password	Password used to authenticate to the iLO. This value is not stored as a custom attribute by MBC.
ilo.reboot_if_on	Default: power the server on only if it is currently off. If you specify this argument with a non-null value, MBC reboots the server, even if it's already on. This value is not stored as a custom attribute by MBC.

The first page of the Web APX has form inputs for the iLO parameters.

The following is an example CSV that will cause MBC to boot/reboot the server:

```
00-13-21-DD-DD-24,pxe_image=linux,sequence_id=310001,  
dhcp.hostname=danube,ilo.hostname=10.128.32.102,  
ilo.username=Administrator,ilo.password=adminpass,  
ilo.reboot_if_on=1
```

Installing (Provisioning) an Operating System

This section describes how to install an operating system on an unprovisioned server using the SA Client, also known as *provisioning* a server.



Before you can perform the tasks described in this section, you or your SA Administrator must have set up your system for OS Provisioning as described in the *SA Policy Setters Guide*.

In order to install an operating system using the SA Client, you must create, define, and run an *OS Sequence* or an *OS Build Plan*. For information about creating OS Build Plans, see [Creating OS Build Plans](#) on page 90. For information about creating an OS Sequence, see [Creating OS Sequences](#) on page 100.

Your SA Administrator must have created the required OS Installation Profile(s) and granted you the required privileges for accessing OS Installation Profile(s) and for OS Provisioning.

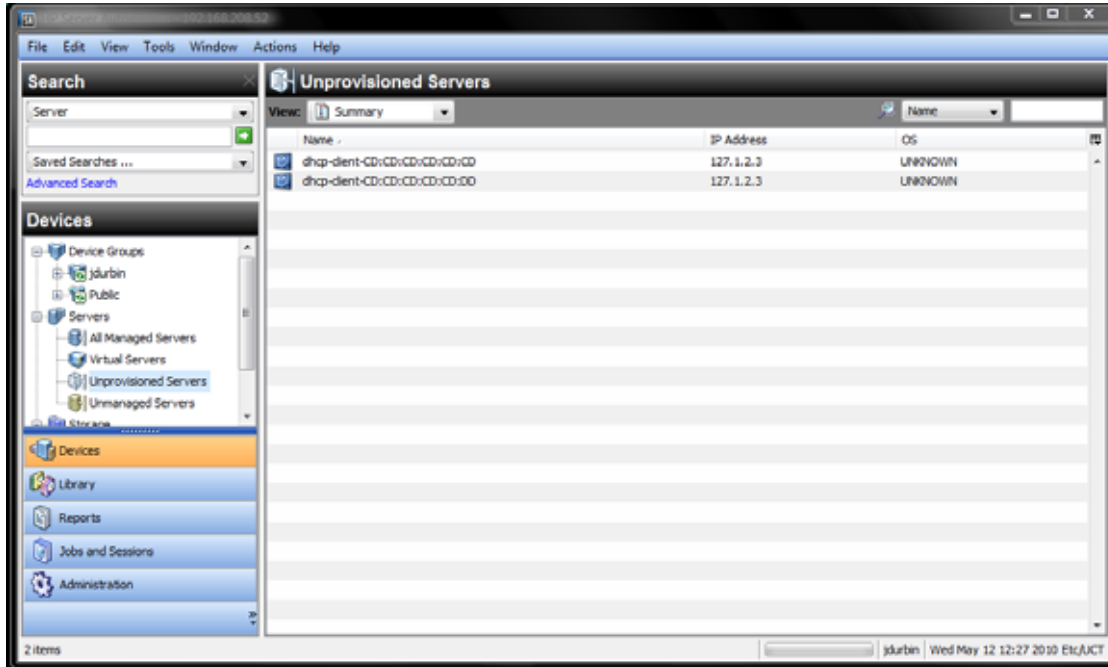
The Unprovisioned Servers List

To provision a server and install an operating system, select an unprovisioned server from the Unprovisioned Servers list in the SA Client. Servers in the Unprovisioned Servers list have registered their presence, but do not have an operating system installed. From this location, you can install an operating system by selecting an unprovisioned server. See [Figure 33](#).



For server records booted using an OGFS Agent to the Unprovisioned Servers list, only OS Build Plan-based provisioning is possible. For server records booted using a Build Agent, only OS Sequence-based provision is possible.


Figure 33 SA Client Unprovisioned Servers List



Select an unprovisioned server in the list and the Content pane will display detailed information about the unprovisioned server that was gathered by the OS Build Agent after a network boot.

The View drop-down list enables you to view the server in the following ways:

- **Summary:** Provides information about the host name set by booting the server the first time over the network or by using an SA Boot CD. It also displays the operating system of the loaded OS Build Agent (Windows, Red Hat Linux, or Solaris), processor type, manufacturer and model of the server, and SA registration information.
- **Properties:** Displays placeholders for various management and reported information which will be filled in later once the server is provisioned.
- **Hardware:** Displays details about the hardware on the server, such as a processor type, physical and virtual memory, storage and network interfaces.
- **Custom Attributes:** Allows you to read and manage custom attributes.
- **History:** Indicates the first event associated with the server.


You can also search for an unprovisioned server using the search tool  in the upper right corner of the Content pane. You can choose a filter, then enter text to search for the server.



You also have the option of running an OS Sequence from the Library and then selecting a server or servers as you configure the Run OS Sequence window.



Some servers in the Unprovisioned Servers list are in a server lifecycle state called *Planned*, which means the server has been partially prepared for OS Provisioning. (It has a device record created for it, but no SA OS Build Agent installed yet.) An OS Sequence can't be run on Servers in the Planned state.

To display the server lifecycle stage value in the Unprovisioned Servers list, in the upper right corner of the Content pane, select the column selector  and from the list select Lifecycle. For more information, see your SA administrator.

Using an OS Build Plan for OS Provisioning

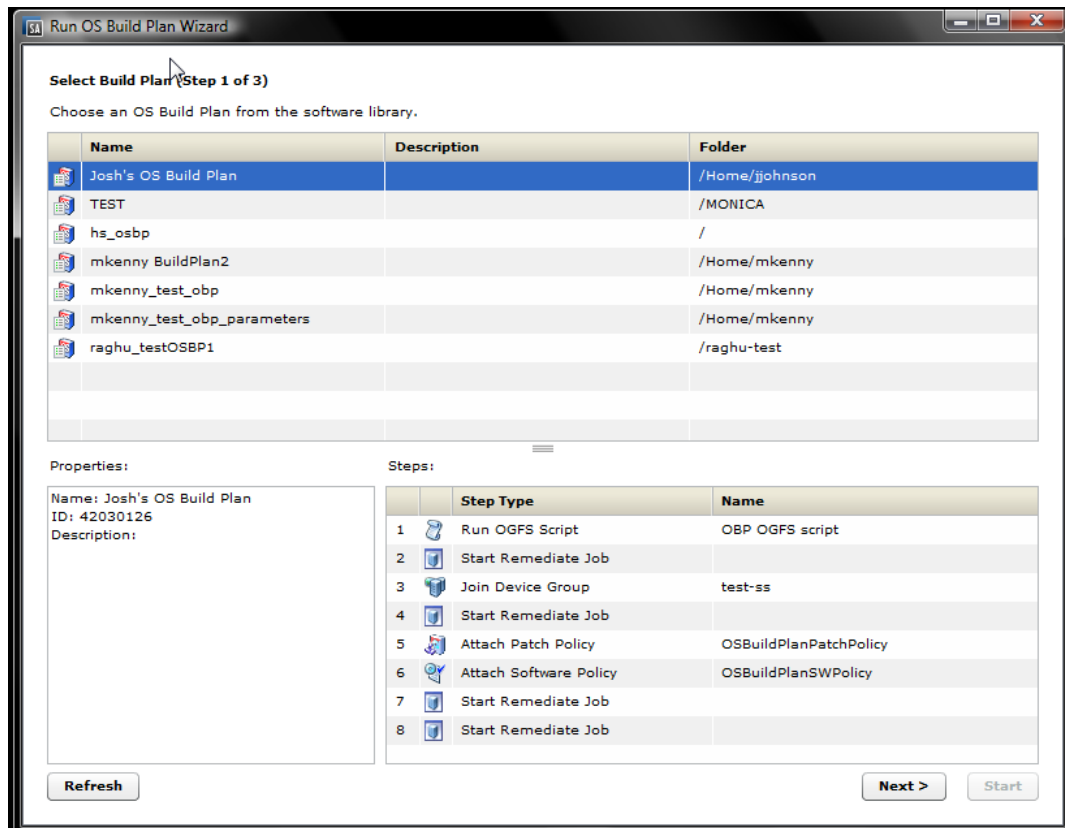
▶ For operating system-specific OS Build Plan usage instructions, see the SA Online Help, OS Provisioning section. This help section also contains information specific to using network parameterization when provisioning VMware ESXi 4.1.

For information about creating OS Build Plans, see [Creating OS Build Plans](#) on page 90.

To install an operating system on an unprovisioned server using an OS Build Plan, perform the following tasks:

- 1 Log on to the SA Client specifying the SA Core that manages the server you will install the operating system on.
- 2 In the Navigation pane, select Devices and select Unprovisioned Servers.
- 3 Select a server from the list of available unprovisioned servers list
- 4 Right-click on a server and select **Run OS Build Plan**. Alternatively, you can select **Action Menu > Run > OS Build Plan** and choose a target server in the Run OS Build Plan window or use the search pane to search for a list of Build Plans.
- 5 SA displays the first page of three of the Run OS Build Plan Wizard.

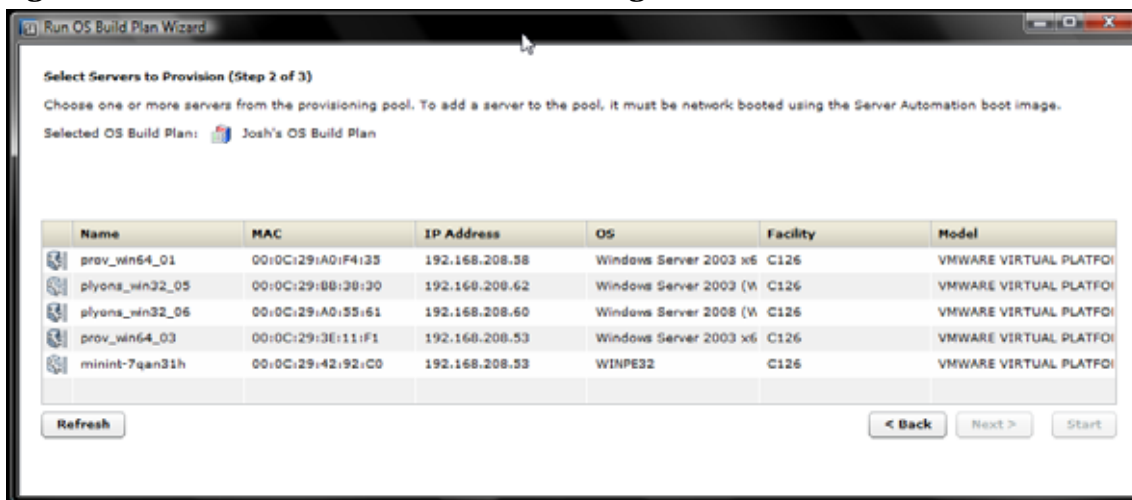
Figure 34 The Run OS Build Plan Wizard - Page 1



On this page you select the OS Build Plan to run. The Properties pane provides a description of the Build Plan, the Steps pane displays the tasks assigned to the Build Plan that will be performed during OS Provisioning.

- 6 Select the OS Build Plan to Run and click Next.
- 7 The second page of the Run OS Build Plan Wizard displays.

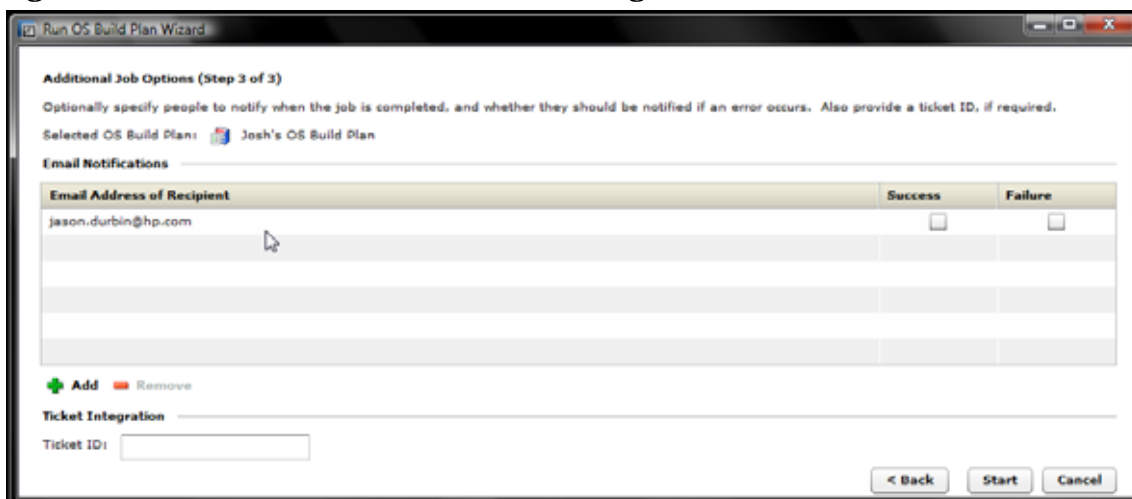
Figure 35 The Run OS Build Plan Wizard - Page 2



On this page you select the server(s) you want to run the OS Build Plan on. If you select a server that is already under SA management, SA displays a warning message that all data on that server will be erased if the Build Plan is run against it.

- 8 Select the server(s) to run the OS Build Plan on and click Next.
- 9 The third page of the Run OS Build Plan Wizard displays.

Figure 36 The Run OS Build Plan Wizard - Page 3



On this page you select email notification options and, if required, specify a Job Ticket ID. Click the green plus icon to add a notification and specify the email address to be notified in case of job failure, success, or both. Highlight an email address and click the red minus icon to remove it. To specify a Job Ticket ID, specify the ID in the Ticket ID field.

- 10 Click Start to begin running the OS Build Plan Job. When the job begins to run, click on the Job in the Job Status window or click **Close** to exit the Job Status window. You can also check the status of the Job by clicking on Job Logs under Jobs and Sessions in Navigation Pane.
- 11 When the OS Build Plan job completes successfully, you can check the **Devices ► All Managed Servers** list to see the newly provisioned server.

Using an OS Sequence for OS Provisioning

An OS Installation Profile is used with OS Sequences and defines all necessary parameters of an operating system, including the operating system type and version, the OS Media Resource Locator (MRL), configuration or response file(s), build customization scripts, and packages required for operating system installation.

OS Installation Profiles are typically defined by an SA Administrator and made available to authorized users for application against servers in SA server pools.

For more information about defining OS Installation Profiles, see [Defining and Managing OS Installation Profiles](#) on page 58.

For information about creating an OS Sequence, see [Creating OS Sequences](#) on page 100.

To install an operating system using an OS Sequence, perform the following tasks:

- 1 Log on to the SA Client specifying the SA Core that manages the server on which you are installing an operating system.
- 2 In the SA Client Navigation pane, select **Devices ► Unprovisioned Servers**.
- 3 Select a server from the list of available unprovisioned servers list
- 4 Right-click on a server and select **Run OS Sequence**.

or


Select **Action Menu ► Run ► OS Sequence** and choose a target server in the Run OS Sequence window.

or



From the Navigation pane, select **Library ► OS Sequences**. Select a folder that contains existing OS Sequences for the correct operating system then right-click an available OS Sequence and select **Run...**




If the **Run OS Sequence** menu item is grayed out, one or more of the unprovisioned servers is in a server lifecycle stage of Planned or you do not have the required permissions to run OS Sequences. Servers in the Planned stage cannot be provisioned. You can display the server lifecycle stage value in the Unprovisioned Servers list.

In the upper right corner of the Content pane, select the column selector . From the list, select **Lifecycle**. For more information, see your SA administrator.

- 5 In the Select OS Sequence pane, click **Add** to add an OS Sequence or click **Next** if OS Sequence is already listed.
- 6 In the Run OS Sequence window, step one requires that you add an unprovisioned server or servers to provision. To add a server, click **Add**.
- 7 Click **Next**, and in the Scheduling pane choose if you want to run the OS Sequence, immediately, or at a later date and time.

- 8 Click **Next** and in the Notifications pane, select an email notifier. Click **Add Notifier** and enter an email address.
- 9 You can specify if you want the email to be sent upon success of the OS Sequence job () or failure of the OS Sequence job ()
- 10 The ticket ID field is only used when Professional Services has integrated SA with your change control systems. It should be left blank otherwise.
- 11 Click **Next**, and review the OS Sequence information before you run the job.
- 12 Click **Start Job** to run the OS Sequence. When the OS Sequence job begins to run, click on the Job in the Job Status window or click **Close** to exit the Job Status window. You can also check the status of the Job by clicking on Job Logs under Jobs and Sessions in Navigation Pane.
- 13 When the OS Sequence job has completed successfully, you can check the **Devices > All Managed Servers** list to see the newly provisioned server.

 If you scheduled the OS Sequence job to run at a later date and would like to cancel it, from the Navigation pane, select **Jobs and Sessions > Recurring Schedules**. Then, select the job, right-click and select **Stop**.

 If an OS Sequence does not have remediation enabled, a newly provisioned servers will not immediately perform a full software registration. Full software registration occurs after a small variable delay, usually less than one hour. Thus when provisioning without remediation, the server's installed software packages and patches might not be listed immediately.

HP-UX Provisioning

HP-UX Provisioning installs HP-UX on bare metal systems using custom configurations. The HP-UX Provisioning process requires the following tasks:

- Setting up an Ignite environment on SA cores/satellites
- Creating custom configurations using the Custom Configuration Editor APX on the SA core
- Booting the target to the Network boot prompt
- Provisioning targets using the HP-UX Provisioning APX on the SA Core

The following sections discuss these tasks.

Prerequisites

You must insure that the following prerequisites are met before you can provision HP-UX servers.

DHCP Setup

To provision servers with unattended installations, DHCP must be enabled. HPSA ships the DHCP server plus a highly customized configuration file to handle DHCP for OS provisioning purposes. To provision HP-UX instead of Linux on Integrity servers, the HP-UX provisioning feature must be enabled on each Core/Satellite.

To enable the HP-UX Provisioning feature, make the following changes on the SA Core:

- 1 **Modify the `/etc/opt/opsware/dhcpd/dhcpd.conf` file by commenting the following line:**

```
#filename "elilo.efi";
```

- 2 **Uncomment the following line to enable network boot for HP-UX Provisioning:**

```
filename "/opt/ignite/boot/nbp.efi";
```

- 3 **Restart the DHCP service:**

```
/etc/opt/opsware/startup/dhcpd restart
```

Ignite Setup on the SA Core

To provision HP-UX servers, you must set up Ignite Configurations. This includes:

- updating the configuration file
- updating the Index file
- copying the golden image archives on each SA core

For detailed information on Ignite-LUX software on a server running Linux, see

<http://www.hp.com/go/ignite-ux-docs>

The following tasks are required to set up the 11.31 golden archive-based configuration on the SA Core:

- **Copy over the sample golden image to the following location:**

```
/var/opt/ignite/archives/B.11.31/
```

- **Copy over the corresponding `.cfg` file to following location:**

```
/var/opt/ignite/data/Rel_B.11.31/B.11.31_archive_IA.cfg
```

- **Add an entry in `/var/opt/ignite/INDEX` for the configuration as follows:**

```
cfg "HP-UX B.11.31 Opsware Archive" {
description "This selection supplies the sample golden archive
created by the IUX team"
"/opt/ignite/data/Rel_B.11.31/config"
"/opt/ignite/data/Rel_B.11.31/hw_patches_cfg"
"/var/opt/ignite/data/Rel_B.11.31/B.11.31_archive_IA.cfg"
"/var/opt/ignite/data/config.local"
}
```

APXs

- SA installs the HP-UX Provisioning APX (Automation Platform Extensions) and Custom Configuration Editor APX which perform parts of the provisioning process. These APXs appear in the SA Client APX Library.

- You can access APXs either through the SA Client or through an SA supported browser. HP recommends running the Custom Configuration Editor APX with Internet Explorer.
- Adobe Flash Player Version 10.0 or above must be installed on all machines on which you plan to run HP-UX Provisioning APXs.

Customer Configuration Subfolders

SA Administrators for HP-UX Provisioning or any user who has privileges to the following folder must create a sample configuration for every customer for whom users want to create configurations:

Library ► By Folder ► Opware ► Tools ► OS Provisioning ► HP-UX Provisioning

The sample configuration is the same as the configuration that is created when you use the Custom Configuration Editor APX. It is called the sample configuration because it is the first configuration created for each new customer. When the first configuration is created, a subfolder is created for that new customer. If the SA Administrator wants to assign restricted access to a user/group based on configurations belonging to a specific customer, they must grant permission to that customer subfolder.

The SA Administrator can see subfolders created with the customer name only after creating the sample configuration.

When you create the sample configuration, make sure that you select the new customer so that the subfolder with the customer name is created immediately within the configuration folder. The SA Administrator can assign read/write access to the user/group to access configurations. For example, say that `PROV_USR` needs access to HP-UX Provisioning and should have access only to configurations belonging to `CustA` and `CustB` customers:

- 1 Open the Custom Configuration Editor APX using Internet Explorer.
- 2 Log on as SA Administrator or as any user who has access to **Library ► By Folder ► Opware ► Tools ► OS Provisioning ► HP-UX Provisioning**
- 3 Create a sample configuration for `CustA` and `CustB` using the Custom Configuration Editor APX.
- 4 Log on to the SA client as SA Administrator for HP-UX Provisioning. Create subfolders named `CustA` and `CustB` at the following location:

Library ► By Folder ► Opware ► Tools ► OS Provisioning ► HP-UX Provisioning/CustA

Library ► By Folder ► Opware ► Tools ► OS Provisioning ► HP-UX Provisioning/CustB

Permissions

This section discusses the minimum permissions required to use the HP-UX Provisioning feature. Your SA Administrators for HP-UX Provisioning can optionally provide additional permissions that enable more features.

User/Group Permissions

SA Administrators for HP-UX Provisioning must grant the following permissions to the user/group:

- Facilities – You must have read/write access to any facility where the Integrity servers are provisioned with the configurations created by the Custom Configuration Editor APX.

- Customers – You must have read/write access to any Not Assigned customer to run the provisioning job successfully.
- You must also have read/write access to any customer on whose behalf the HP-UX configurations are created.
- Features – You must have Managed Server and Groups permission so that you can actually see the server in SA after you provision it.

Folder Permissions

SA Administrators for HP-UX Provisioning must also grant folder permissions to list APXs, software policies, and configurations.

- APXs – You must have List Contents Of Folder and Execute Objects Within Folder permissions to the following folder to access HP-UX Provisioning and Custom Configuration Editor APXs:
Library > By Folder > Opware > Tools > OS Provisioning > HP-UX
- Software Policies – You must have List contents of Folder, Read Objects Within Folder, and Execute Objects Within Folder permissions to the folders where HP-UX software policies used to define the configurations are placed.
- Configurations – You must have Read Objects Within Folder and Write Objects Within Folder permissions to the following folder because it contains the HP-UX configurations:
Library > By Folder > Opware > Tools > OS Provisioning > HP-UX Provisioning/<customer_name>

Installing an Operating System on HP-UX Servers

HP-UX Provisioning installs HP-UX on bare metal systems using custom configurations. The HP-UX Provisioning process requires the following tasks:

- Setting up an Ignite environment on SA cores/satellites
- Creating custom configurations using the Custom Configuration Editor APX on the SA core
- Booting the target to the Network boot prompt
- Provisioning targets using the HP-UX Provisioning APX on the SA Core

The following sections discuss these tasks.



You must have set up Ignite before continuing. For more information, see [Ignite Setup on the SA Core](#) on page 131.

Creating a Custom Configuration

You can specify customized configurations to be applied to an Integrity server. You can specify Ignite attributes that are applied on the server during HP-UX installation on top of the standard golden image configurations. You can also select additional software policies to be remediated as part of the HP-UX installation.

You can build customer-specific configurations by specifying the platform, base configuration, Ignite attributes, and related software policies. You can customize the installation to meet your specific needs.

To provision the server in a customized way, you must first create a custom configuration.

HP-UX Custom Configuration Editor APX

To access the HP-UX Custom Configuration Editor APX, perform these tasks:

- 1 In the SA Web Client: Open Internet Explorer and specify the URL:

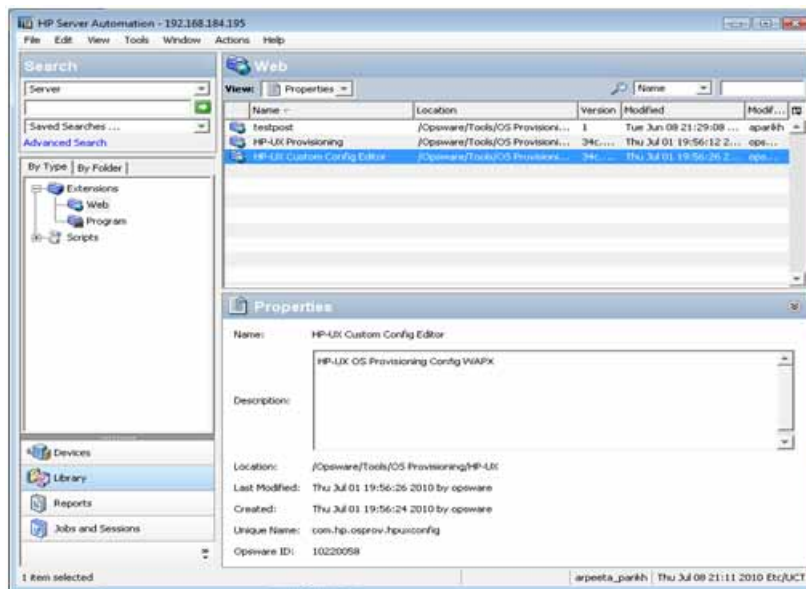
https://<your core>/webapp/<APX Unique Name>

The unique name for the APX is displayed in the SA Web Client.

- 2 In the SA Client: Click the **Library** tab and select **Extensions** ► **Web** ► **HPUX Custom Config Editor**. The unique name for the APX is displayed when you select the APX.

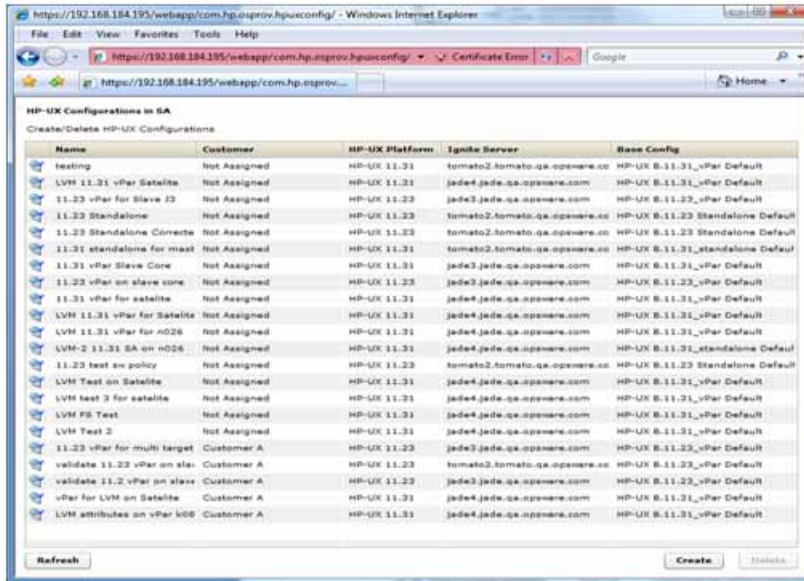
In this instance, the Custom Configuration Editor APX name is `com.hp.osprov.hpuxconfig`.

Figure 37 HP-UX Custom Configuration Editor APX



All existing custom configurations are listed with Name, Customer, HP-UX Platform, Ignite Server, and Base Config details. The Refresh, Create, and Delete buttons also appear.

Figure 38 HP-UX Custom Configuration Editor APX - List Existing Configurations



The screenshot shows a web browser window displaying the HP-UX Custom Configuration Editor APX interface. The page title is "HP-UX Configurations in SA". Below the title, there is a table listing various configurations. The table has five columns: Name, Customer, HP-UX Platform, Ignite Server, and Base Config. The configurations listed include "testing", "LVM 11.31 vPar Satellite", "11.23 vPar for Slave J3", "11.23 Standalone", "11.23 Standalone Corrects", "11.31 standalone for maat", "11.31 vPar Slave Core", "11.23 vPar on slave core", "11.31 vPar for satellite", "LVM 11.31 vPar for Satellite", "LVM 11.31 vPar for n026", "11.23 test sw policy", "LVM Test on Satellite", "LVM test 3 for satellite", "LVM FS Test", "LVM Test 2", "11.23 vPar for multi target", "validate 11.23 vPar on sla", "validate 11.2 vPar on slave", "vPar for LVM on Satellite", and "LVM attributes on vPar k00". The "Customer" column shows "Not Assigned" for most configurations and "Customer A" for the last four. The "HP-UX Platform" column shows "HP-UX 11.31" and "HP-UX 11.23". The "Ignite Server" column shows "tomato2.tomato.qe.opsware.co" and "jade3.jade.qe.opsware.com". The "Base Config" column shows various default configurations like "HP-UX B.11.31_vPar Default", "HP-UX B.11.23 Standalone Default", and "HP-UX B.11.31_standalone Defaul". At the bottom of the table, there are "Refresh", "Create", and "Delete" buttons.

Name	Customer	HP-UX Platform	Ignite Server	Base Config
testing	Not Assigned	HP-UX 11.31	tomato2.tomato.qe.opsware.co	HP-UX B.11.31_vPar Default
LVM 11.31 vPar Satellite	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
11.23 vPar for Slave J3	Not Assigned	HP-UX 11.23	jade3.jade.qe.opsware.com	HP-UX B.11.23_vPar Default
11.23 Standalone	Not Assigned	HP-UX 11.23	tomato2.tomato.qe.opsware.co	HP-UX B.11.23 Standalone Default
11.23 Standalone Corrects	Not Assigned	HP-UX 11.23	tomato2.tomato.qe.opsware.co	HP-UX B.11.23 Standalone Default
11.31 standalone for maat	Not Assigned	HP-UX 11.31	tomato2.tomato.qe.opsware.co	HP-UX B.11.31_standalone Defaul
11.31 vPar Slave Core	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
11.23 vPar on slave core	Not Assigned	HP-UX 11.23	jade3.jade.qe.opsware.com	HP-UX B.11.23_vPar Default
11.31 vPar for satellite	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
LVM 11.31 vPar for Satellite	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
LVM 11.31 vPar for n026	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
LVM-2 11.31 SA on n026	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_standalone Defaul
11.23 test sw policy	Not Assigned	HP-UX 11.23	tomato2.tomato.qe.opsware.co	HP-UX B.11.23 Standalone Default
LVM Test on Satellite	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
LVM test 3 for satellite	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
LVM FS Test	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
LVM Test 2	Not Assigned	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
11.23 vPar for multi target	Customer A	HP-UX 11.23	jade3.jade.qe.opsware.com	HP-UX B.11.23_vPar Default
validate 11.23 vPar on sla	Customer A	HP-UX 11.23	tomato2.tomato.qe.opsware.co	HP-UX B.11.23_vPar Default
validate 11.2 vPar on slave	Customer A	HP-UX 11.23	jade3.jade.qe.opsware.com	HP-UX B.11.23_vPar Default
vPar for LVM on Satellite	Customer A	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default
LVM attributes on vPar k00	Customer A	HP-UX 11.31	jade3.jade.qe.opsware.com	HP-UX B.11.31_vPar Default

When you select any existing configuration from the list, you will see additional details such as Custom Configurations and Software policies.

The Configuration listings are based on permissions assigned to you. You are able to list configurations belonging to the customers for which you have been granted permission. The configuration can be customer dependent or customer independent.

A customer independent configuration is accessible to all users. A customer dependant configuration is accessible to only those users who have appropriate permissions assigned. You can make the configuration dependant or independent by selecting the customer when you create the configuration.

The HP-UX Custom Configuration Editor APX enables you to:

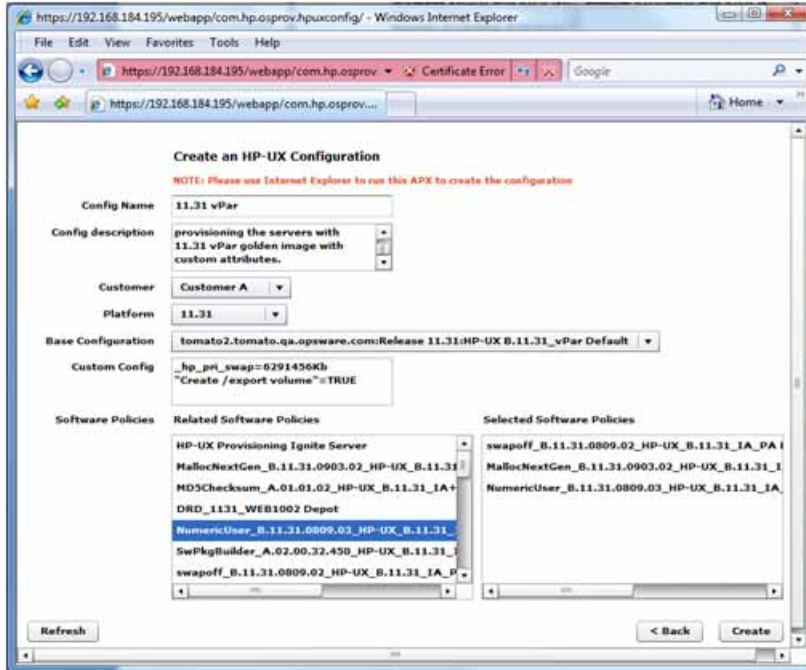
- Create Custom Configurations
- Delete Custom Configurations

Creating a Custom Configuration

To create a custom configuration, follow these steps:

- 1 Open the HP-UX Custom Configuration Editor APX using Internet Explorer.
- 2 Click the Create button and specify the required details to create a new custom configuration.

Figure 39 HP-UX Custom Configuration Editor - Create Custom Config



3 You must specify the following required details to define the HP-UX custom configuration:

- Config Name: This is a mandatory field. It must be unique for each customer. The APX validates the following specifications:
 - Must not exceed 255 characters in length.
 - Must not begin or end with an empty space.
 - Must not begin with punctuation, including @#\$\$%^&*() +_-,./:;~{ } [] | \ ' " ? = `
 - Should not use newline, tab, flash or backslash.
- Config Description: This is an optional field that can contain explanatory text describing the purpose and use of the configuration.
- Customer: By default, this is set to Not Assigned, which makes the configuration customer independent. You can only list those customers for which you have permission. You will not be able to list the configurations if appropriate permissions have not been granted to you for that customer.
- Platform: This is a mandatory field. You must select either 11.31 or 11.23 from the drop-down menu. Base configuration or Related Software Policies are dependent on platform selection. If the platform was not selected and you try to select either Base configuration or Related Software Policies, a warning message appears.
- Base Configuration: Content is displayed based on the HP-UX platform selected.
- Custom Config: This is an optional field that can be used to specify Ignite attributes. Any valid Ignite attributes specified in this field overwrite the configuration values specified in the golden image, which allows you to provision the servers in a customized manner.

The following are examples of Ignite attributes:

```
_hp_pri_swap=6291456Kb
"Create /export volume"=TRUE
```



```
_hp_root_disk="0/1/1/1.2.0"  
_hp_disk_layout="Logical Volume manager (LVM) with VxFS"
```

For detailed information on Ignite custom attributes, see

<http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01942568/c01942568.pdf>

If the attributes and values specified are valid, they are applied on the servers on top of the standard image configurations. However, if the attributes specified are invalid or have incorrect syntax, provisioning will not start.

In some cases, attributes and values are incompatible. For example, say that you want to provision the server with the following Ignite attributes:

```
_hp_pri_swap=6291456Kb  
"Create /export volume"=TRUE
```

The Ignite attributes syntax is correct and also has valid values but your target does not have enough disk space to implement it. As a result, a warning message is displayed on the targets before provisioning starts.

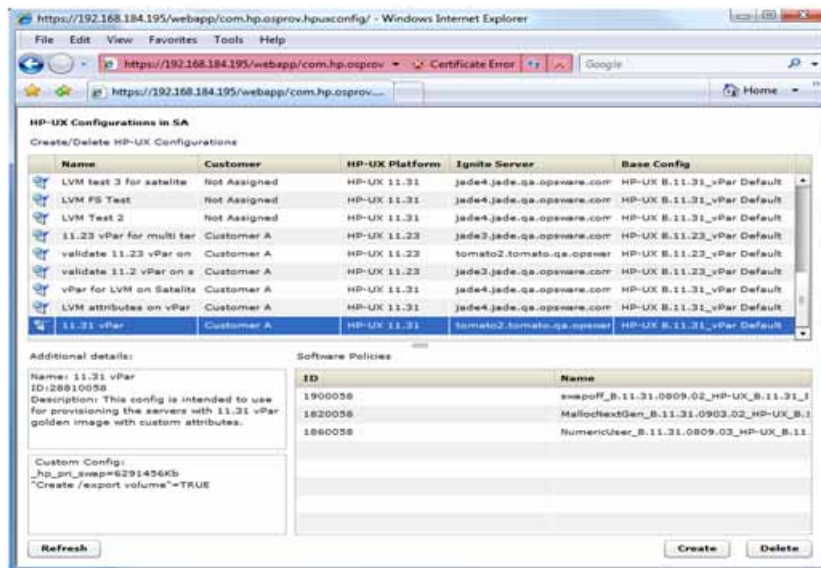
- 4 **Related Software Policies:** This is an optional field consisting of a drop-down menu whose contents are displayed based on the platform selected. You can select multiple Related Software policies by holding the CTRL key and dragging the policies to the selected software policies list to apply them on the server after provisioning is completed and the agent is installed.

You can change the sequence of a selected software policy by dragging it up or down. Policies specified in Selected Software Policies are applied on top of the standard policies in the golden Ignite image.

After the details for all mandatory fields, Config Name, Platform and Base Config are specified, the Create button is enabled.

- 5 Click the Create button. A confirmation message appears and the newly created configuration is listed.

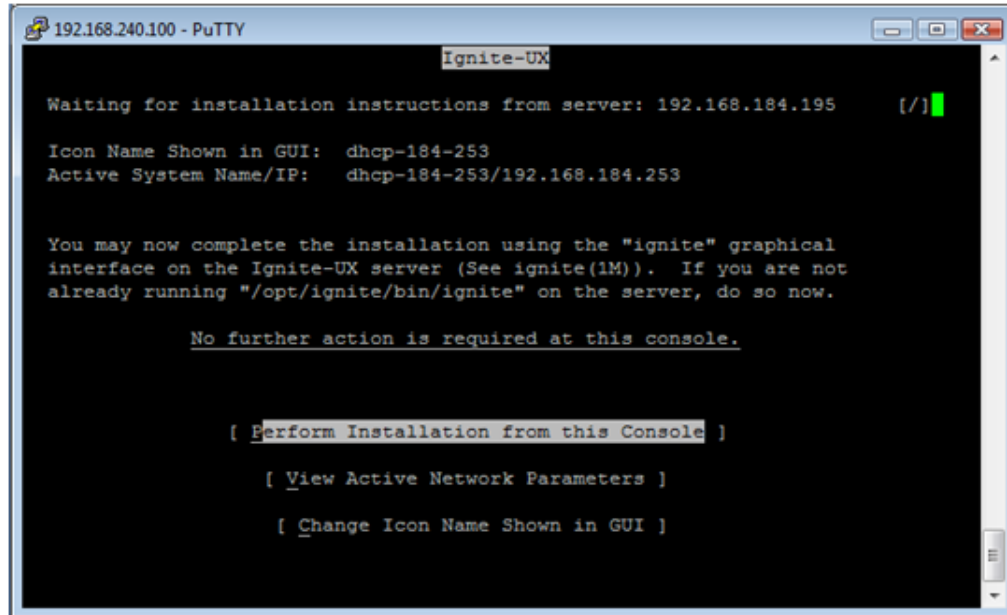
Figure 40 Newly Created Custom Config Profile



Boot Target

The network booted integrity client requires selecting the desired LAN and target OS to install. It waits for server side install instructions to start HP-UX Provisioning. The following figure shows a target client waiting to be installed.

Figure 41 Server Waiting to be Installed



For more information, see:

<http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01868281/c01868281.pdf>

Provision the Target Servers

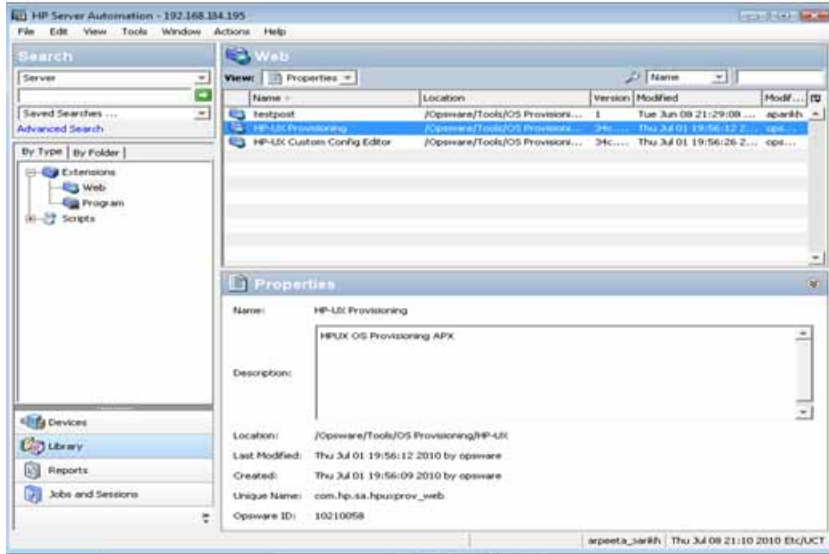
Once the custom configuration is created, it is listed on the HP-UX Provisioning APX on the SA Client. The target server waiting at the network boot prompt is listed under the unprovisioned servers list on the HP-UX Provisioning APX. The following section describes how to provision the targets.

HP-UX Provisioning APX

To access HP-UX Provisioning APX, follow these steps:

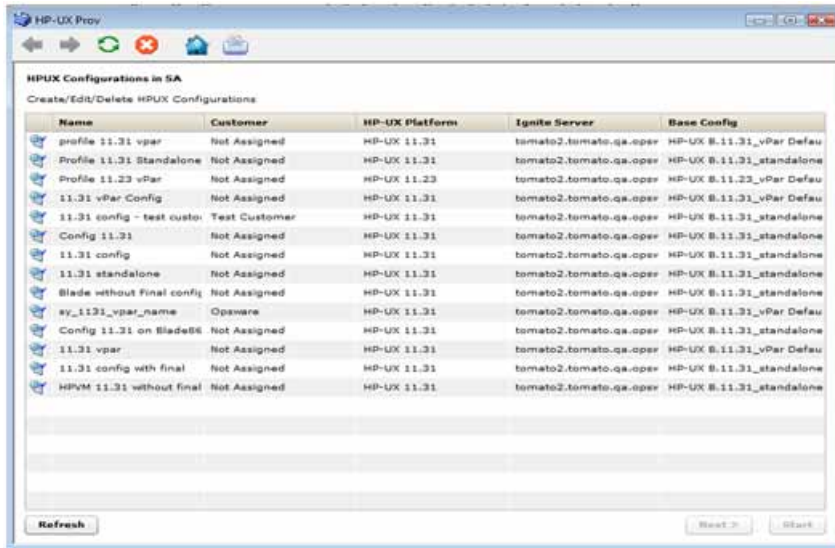
- 1 Log on to the SA Client.
- 2 Click the **Library** tab and select **Extensions** ► **Web** ► **HP-UX Provisioning**.

Figure 42 HP-UX Provisioning APX



The HP-UX Provisioning APX popup window showing the HP-UX configuration opens.

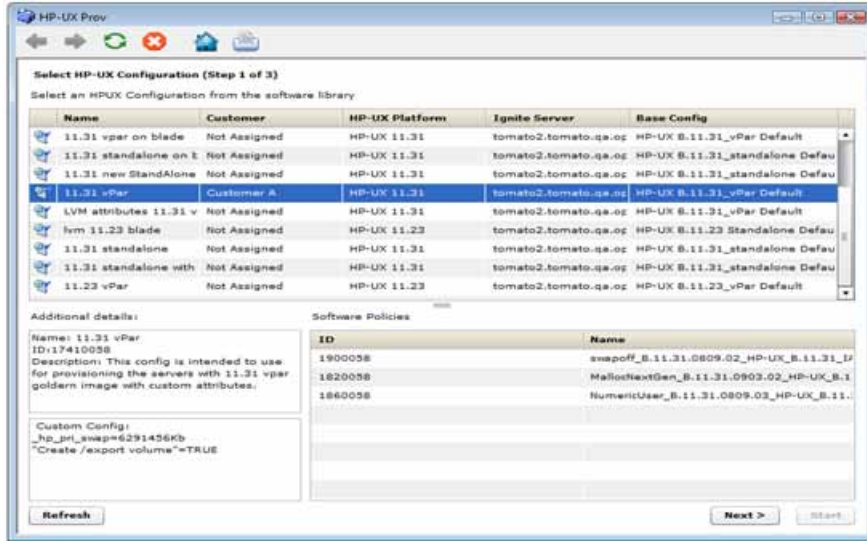
Figure 43 HP-UX Provisioning APX - Listing of HP-UX Configurations



All the configurations created using the Custom Configuration Editor APX are listed on the HP-UX Provisioning APX based on permissions granted. Configurations are listed with Name, Customer, HP-UX Platform, Ignite Server, and Base configurations details. The Refresh, Next, and Start buttons are also displayed. Select the HP-UX configuration you want installed on the servers and click **Next**.

When you select a configuration, additional details, including Custom Configurations and Software policies of the chosen configuration, are displayed. When you select a configuration, the Next button is enabled.

Figure 44 HP-UX Provisioning APX – Select HP-UX Configuration

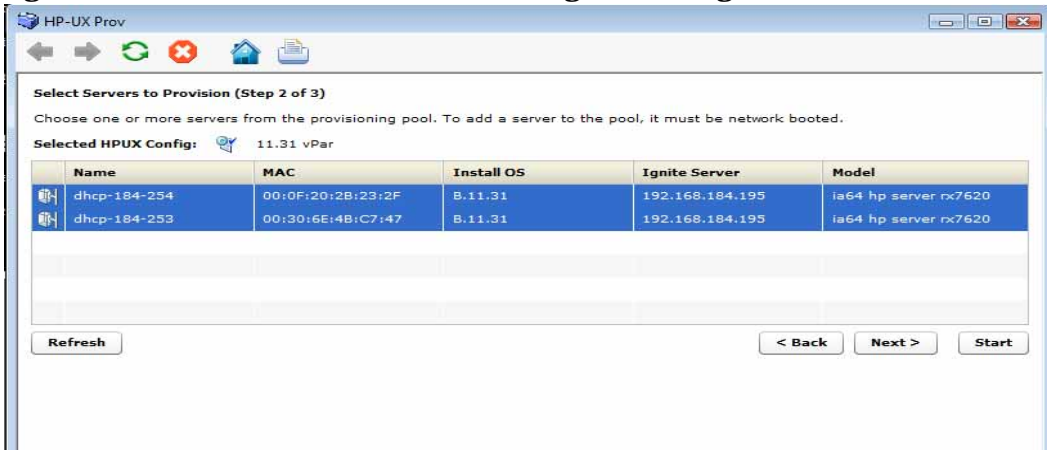


- 3 Click **Next** to select the unprovisioned servers.

All unprovisioned servers waiting at the network boot prompt matching the selected configuration platform are displayed and show MAC address, install OS, Ignite Server and Model details. Servers in the Unprovisioned Servers list register their presence, but do not have an operating system installed.

- 4 Select the server to provision. Hold down the CTRL key to select multiple servers to provision at the same time using the same configuration.

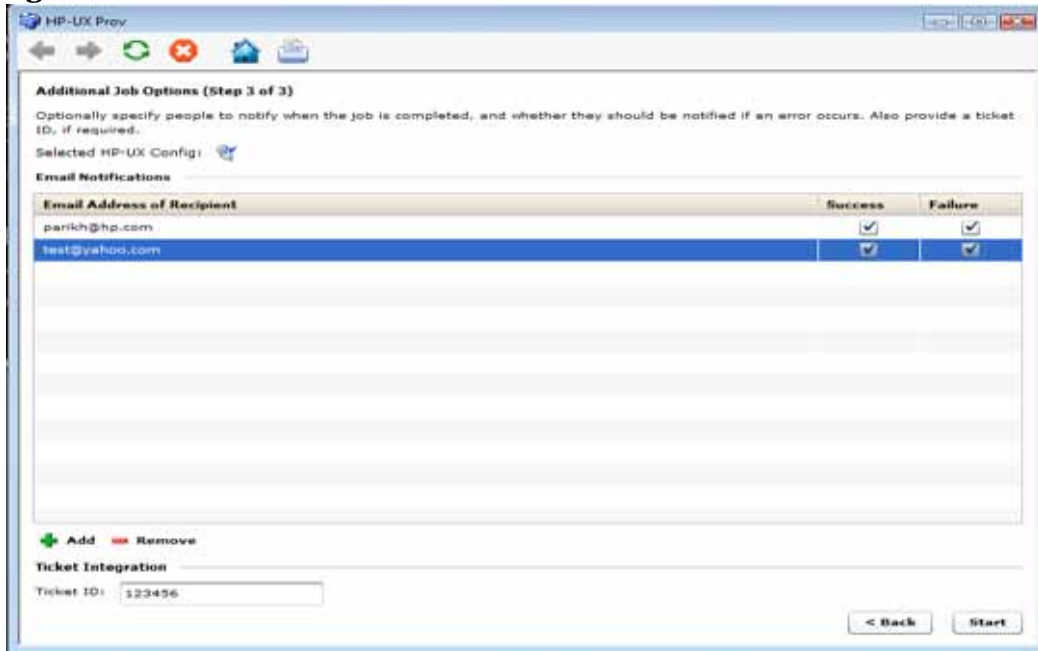
Figure 45 Select Client to Provision Using the Config Selected Before



Once all servers are selected, provisioning starts immediately after you click the Start button.

- 5 To set email notification, click the **Next** button. The following screen appears.

Figure 46 Set the Email Notification



On Email Notification, by default, your email address (the user running the job) is displayed. To add additional email addresses, click **+Add**. Select the check boxes to receive notifications when job failure or success occurs. To remove an email address, select the address and click **Remove**.

You can also specify the Job Ticket ID in the Ticket Integration section. This Ticket ID is associated with the Job.

- 6 When you click Start, the job is initiated for program APX and the Job ID is assigned to it.

Figure 47 Initializing the Provisioning Job

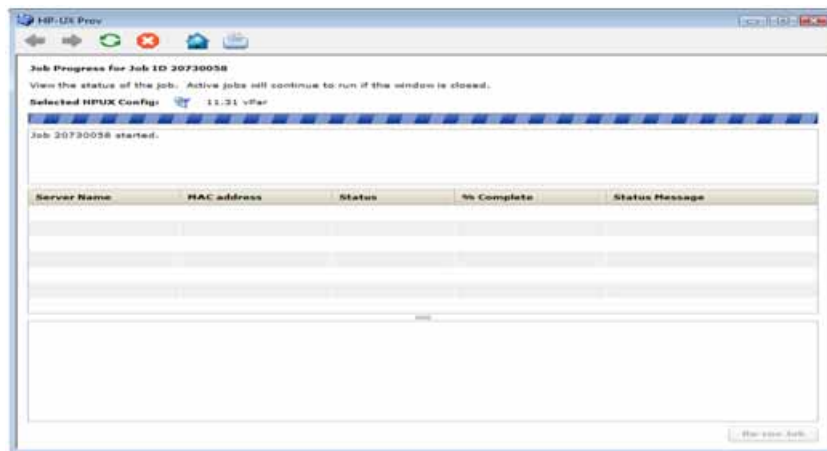
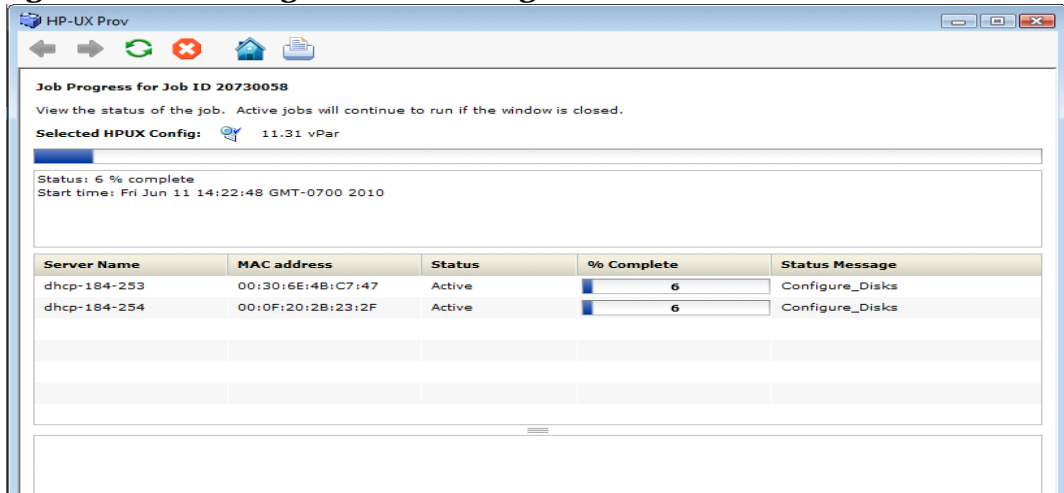


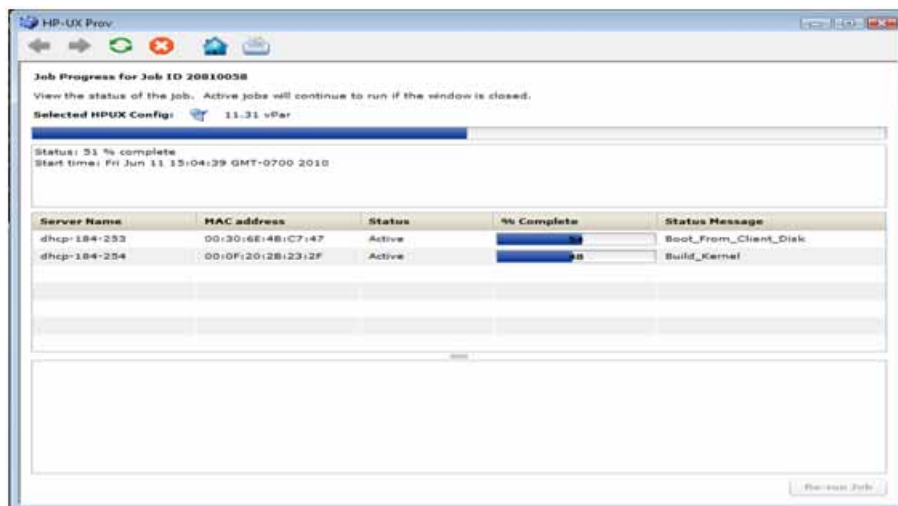
Figure 48 Initializing the Provisioning Job



If the provisioning job was initiated successfully on the servers, you will see the following screen. This screen has a progress bar that is refreshed with updated progress status messages. The following status messages are updated during the provisioning job:

- Waiting_to_install
- Prepare_Config_File
- Configure_Disks
- Download_mini-system
- Loading_software
- Build_Kernel
- Boot_From_Client_Disk
- Run_Postconfigure_Scripts
- Agent Install
- Remediate software policy

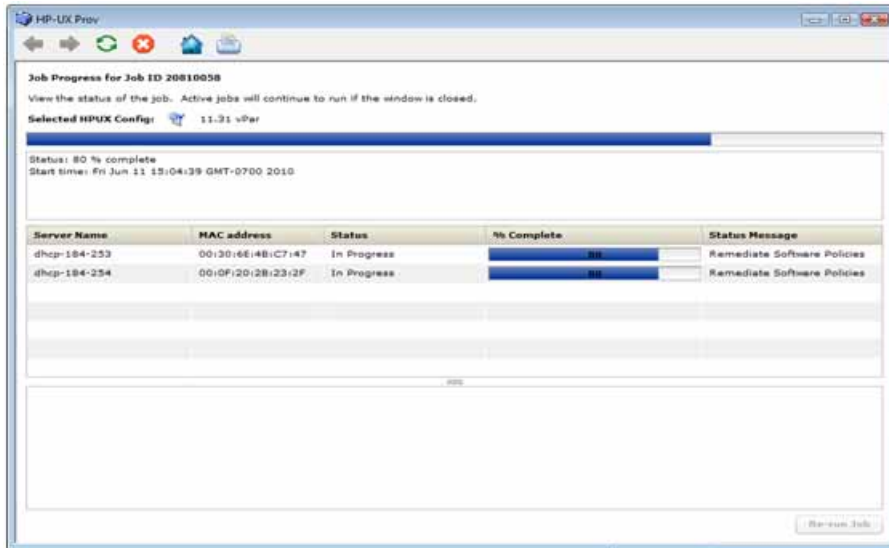
Figure 49 Progress Bar with Status Message



Once the provisioning job starts, two different progress bars are displayed. The consolidated progress bar displays the average percentage of progress on all servers being provisioned. It also displays the average percentage of jobs finished with the job start time.

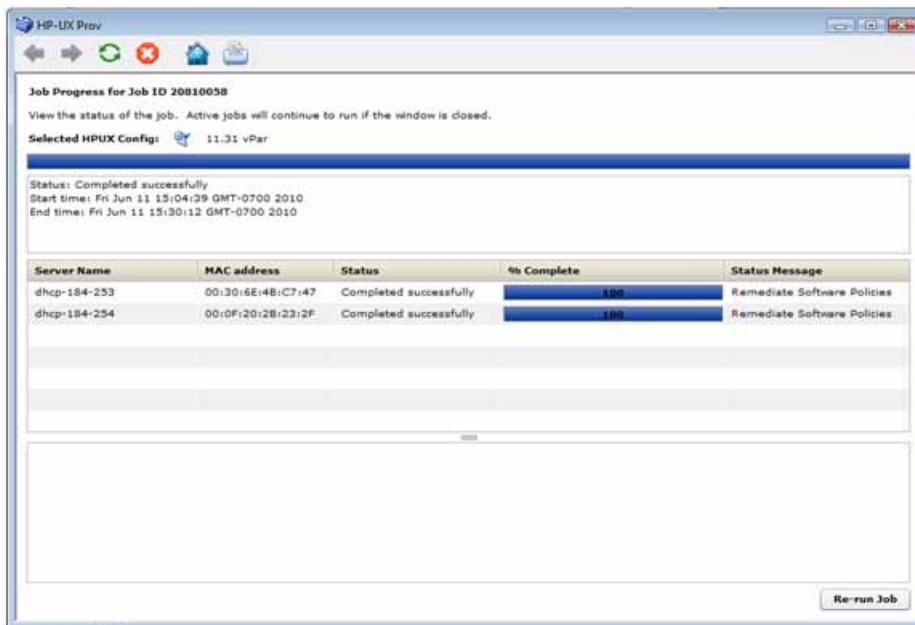
The progress bar for each of the servers being provisioned shows the percentage of provisioning complete with Server Name, MAC address, Status, and Status Message details. The % Complete and Status Message are updated along with the progress of the provisioning job.

Figure 50 Progress Bar with Status Message Remediate Software Policies



Once the server is provisioned, the agent is installed by default. Also, the software policies chosen in the configuration are remediated on the servers. When the HP-UX Provisioning job completes, an email is sent to you if you set up email notification.

Figure 51 Progress Bar with Job Completed Message



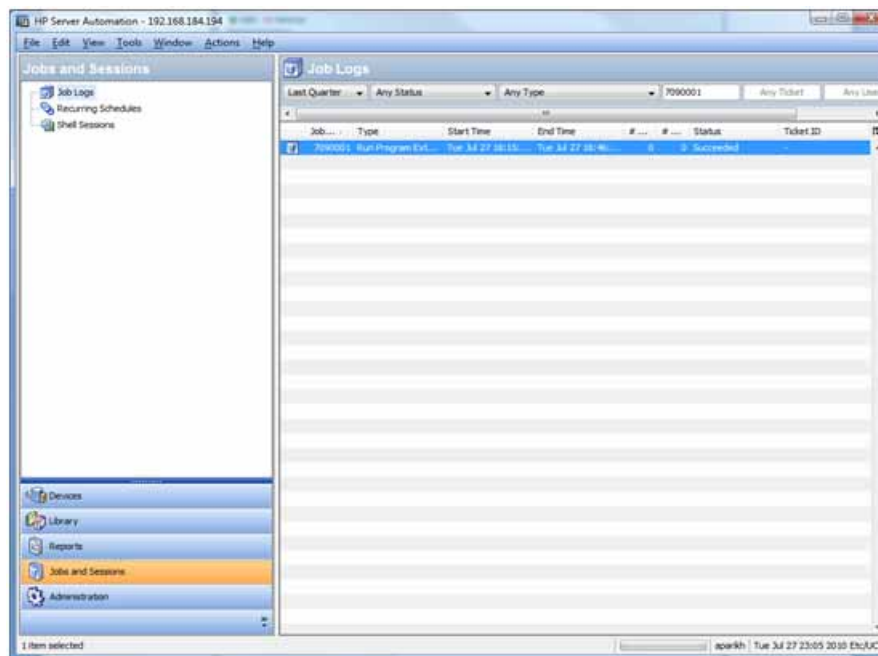
The provisioned servers are managed by SA. The servers are listed under **Devices** ► **All Managed Servers**. Select the server, and then select Go to **View** ► **Properties**. You can see the customer value, which is the same as the configuration customer value. You can see the servers listed as managed servers only if you have permission granted for the customer.

You can also verify the configuration name associated with the server. Go to **View** ► **Custom Attributes**. This can be useful to find out which configuration was used to provision the server.

Jobs and Sessions on SA Core

HP-UX Provisioning APX assigns the job ID. You can use the job ID to verify the job status at the following location: **SA** ► **Jobs and Sessions** ► **Job Logs**.

Figure 52 Job Status on SA Jobs and Sessions



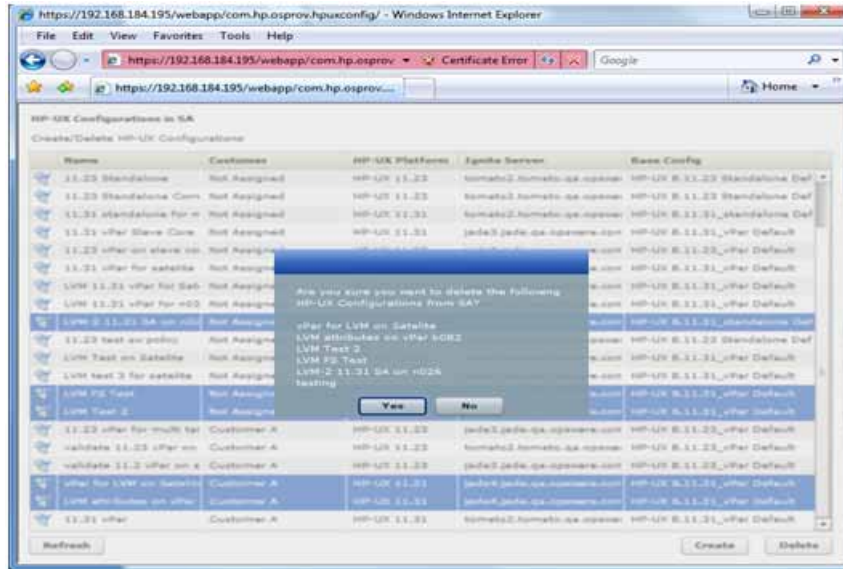
You can also find the specified Job Ticket ID in the notifications tab by double-clicking the job.

Deleting Custom Configurations

To delete a custom configuration, perform the following tasks:

- 1 Open the HP-UX Custom Configuration Editor APX using Internet Explorer.
- 2 Select the configuration or hold the CTRL key to select multiple configurations and click **Delete**.

Figure 53 Delete Custom Config Profile



- 3 Click **Yes** on the confirmation window. The selected configurations are deleted and are no longer listed.

You are allowed to delete only those configurations for which you were granted privileges to execute the delete configuration operation.

Glossary

Ignite-UX

An HP-UX administration toolset that allows:

- simultaneous installation of HP-UX on multiple clients
- creation of custom installation configurations (golden images) for multiple installations on clients
- creation of recovery media
- recovery of HP-UX clients both locally and remotely

Ignite-UX server

A server from which Ignite-UX is used to install HP-UX on client systems.

Golden Image

A combination of a golden archive and a configuration file describing a system's disk layout and file system. Used as a common configuration to install clients.

Ignite Attributes

Custom attributes that allow provisioning of a server with new customized values that overwrite the standard attributes values defined in the golden image.

Network boot

A system boot of the HP-UX install kernel over a network connection from an Ignite-UX server.

Target or Target Server

The HP Integrity server to be provisioned.

Custom Configuration Editor APX

The APX used to create and delete the custom configurations for HP-UX Provisioning.

HP-UX Provisioning APX

The APX used to start HP-UX Provisioning on target servers.

Sample Configuration

The first configuration created for the new customer by the SA Administrator for HP-UX Provisioning. It is same as the custom configuration but is the first configuration for a new customer. It creates a subfolder with the customer name under the HP-UX Configs folder in the SA Client Library.

Useful Links

- 11iv3 installation information:
<http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01916012/c01916012.pdf>
- White paper: *Ignite-LUX: Management and Integration of Ignite-UX Software on a Server Running Linux* at:
<http://www.hp.com/go/ignite-ux-docs>
- Ignite-UX custom configuration files:
<http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01942568/c01942568.pdf>
- Ignite-UX:
<http://h20000.www2.hp.com/bizsupport/TechSupport/DocumentIndex.jsp?lang=en&cc=us&taskId=101&prodClassId=10008&contentType=SupportManual&docIndexId=64255&prodTypeId=18964&prodSeriesId=4077173>

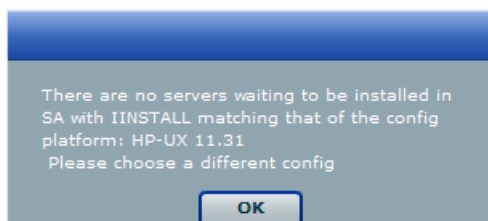
Troubleshooting

Following are some problem scenarios and suggested solutions.

Scenario: No Servers Waiting to be Installed

If there are no servers waiting at the network boot prompt with the HP-UX version that matches the selected configuration's HP-UX version, the following message is displayed:

Figure 54 No Servers Waiting to be Installed



Ensure that you have the selected correct configuration.

Scenario: Servers Waiting to be Installed are Managed Servers

If there are servers waiting for network installation but they are already managed by SA, the following warning message is displayed.

Figure 55 Servers Waiting to be Installed are Managed Servers



This warning message indicates that the listed servers are waiting for installation but are not candidates for reprovisioning because they are listed as Managed Servers in SA. To continue reprovisioning these servers, you must manually delete them from the SA managed server list.

For more information about deactivating and deleting a server from the SA managed servers list, see the *SA Users Guide: Server Automation*.

Figure 56 List of Deleted Managed Servers for Reprovisioning

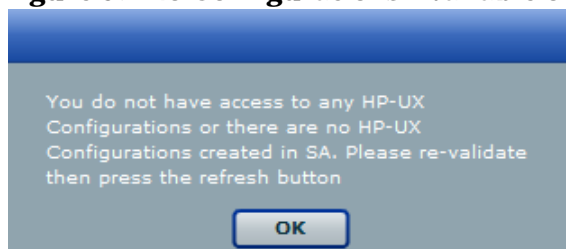


Once a server is deleted, it is not listed under the SA managed server list. Click **Refresh** in the HP-UX Provisioning APX window and the server should be listed under the unprovisioned server pool. Select the server and continue provisioning it.

Scenario: Configurations Unavailable or Permissions Not Granted

This message appears when you do not have enough permission granted to list the configurations or there are no configurations found.

Figure 57 No Configurations Available or Permissions Granted



Contact your SA Administrator to obtain permission or create required configurations using the Custom Configuration Editor APX.

Scenario: Incorrect Target Listing

In certain error scenarios, you may see stale data in the APX client's menu such as clients that are not currently waiting to be network installed or clients with an incorrect hostname.

- A client that is not currently waiting to be network installed is displayed in the APX clients list.

If the target server is reset while waiting to be network installed, the Ignite-UX cannot detect the change and does not update the client's status.

Retry the installation or delete the directories for the target under `/var/opt/ignite/clients/`. There are two directories for each client, one of the form `<mac address>` (for example, `0x00306EF37245`) and the other a symbolic link to the directory. Delete both directories.

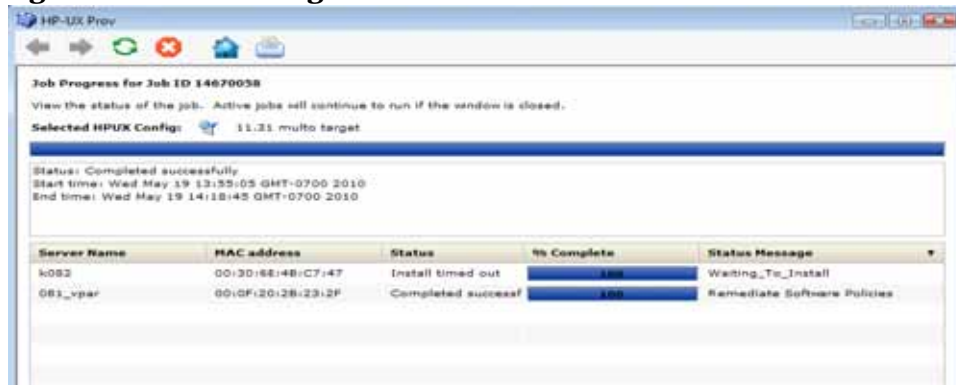
- A client is listed in the APX with an incorrect hostname.

This can happen when you modify DHCP to provide a different hostname after having previously provisioned the client. Ignite UX reuses the directories in `/var/opt/ignite/clients/` it set up for a client (when it finds a client based on the MAC address), so the APX reuses that information. You can delete the two directories for the client under `/var/opt/ignite/clients` and retry the installation.

Scenario: Installation Timed Out Error

An installation timed out error occurs when the provisioning job is not initiated on the target server. This could be due to a network issue, the golden image not being available, or for other reasons.

Figure 58 Provisioning Not Initiated on Server



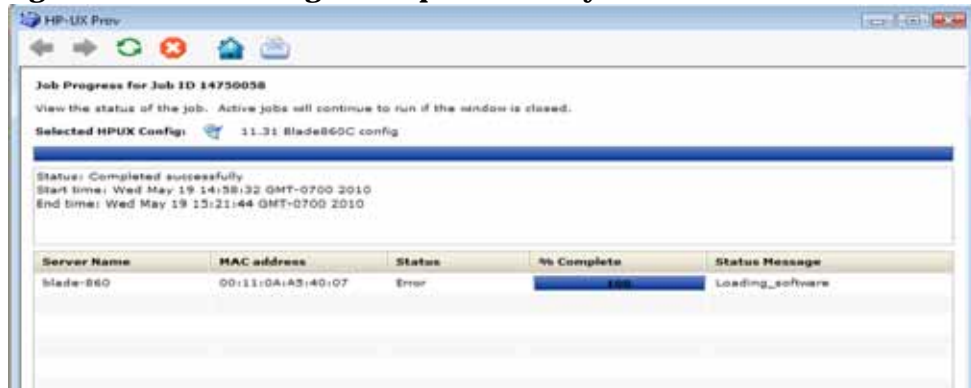
Ensure that the network connection and Ignite images are accessibility and run the APX again to initiate provisioning.

Scenario: Loading Software Error

A loading software error can occur due to:

- Network issues
- Corresponding archive missing or not accessible
- Incorrect setup of golden images

Figure 59 Golden Image Set Up Incorrectly

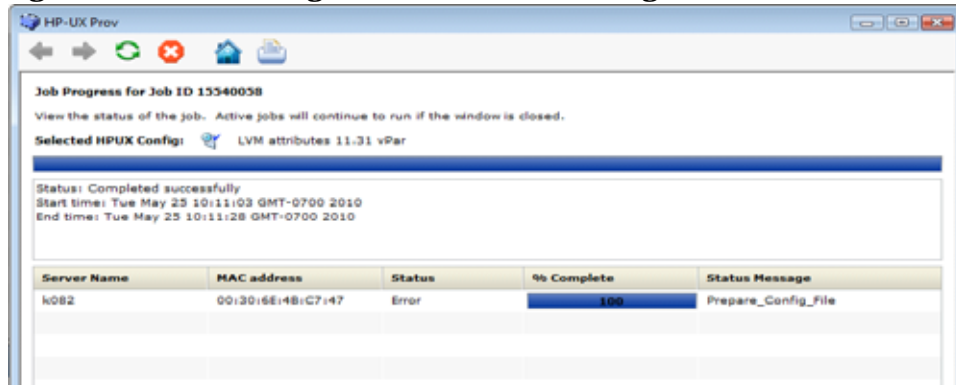


To resolve this, ensure that the Ignite-specific configuration file, Index file, and archives are correctly set up and pointing to correct locations. Also ensure that the network connection between the target and the Ignite server is accessible.

Scenario: Prepare Config File Error

The provisioning job fails to initiate on the server when any syntax errors are found in the custom attributes specified in the configuration or when the custom attributes are not compatible.

Figure 60 Provisioning Failed Due to Invalid Ignite Attributes



You may need to reboot the system and bring it back to the network boot prompt, then create a new configuration with corrected custom attributes. Ensure that the specified syntax is correct and compatible.

Scenario: Agent Fails to Start

If, after successful job completion, the SA Agent fails to start on a newly provisioned target, the golden image you used may already have an Agent installed.

For example, as part of the standard provisioning process, after HP-UX is installed on the server, a post-install script that installs an Agent runs on the server. Because the Agent was previously installed with the golden image, the Agent may not start.

Model Base Packages Functionality

OS Provisioning provides the ability to create software policies that model the base set of packages installed during OS Provisioning.

During OS Provisioning — after the base operating system installation, agent installation, and reachability test, but before reconcile/remediate — a new script triggers software registration on the newly provisioned server, then models the installed packages as a software policy.

To activate this functionality, the server being provisioned must have a custom attribute defined (or inherited) named `model_base_packages`. The value for this attribute must either be empty or an absolute folder path to the name of the software policy to be created (or updated) with the package list.

If the `model_base_packages` value is empty, a software policy is created (or updated if it already exists) in the same folder as the OS Sequence. The software policy name will be the OS Sequence name plus `Base Packages`.

Each installed package that is successfully found in SA is added to the list of software policy items. A list of package names and versions that were not found in SA will be available as a custom attribute named `missing_packages` in the software policy. This policy is attached to the OS Sequence which has remediation enabled. Because the above occurs before remediation, this policy is included in the remediation, thus adopting the modeled packages since they are by definition already installed.

You should only specify the `model_base_packages` custom attribute value as empty when running OS Sequences from the SA Client. When running OS Provisioning from the SA Web Client, the `model_base_packages` custom attribute value must be the path to the Software Policy.

The only valid value for the `model_base_packages` custom attribute is the path to a Software Policy. For example:

```
/Customer/OS Baselines/Solaris 10 baseline Q4 2007
```

In this case, the Software Policy will be created at the specified path and with the specified name. Any folders that are missing will automatically be created. If the Software Policy already exists, it will be updated.



When run from the SA Web Client Install OS wizard, the Software Policy will be attached to the server being provisioned. However, since the Install OS wizard triggers a legacy reconcile, remediate is bypassed so the policy will not be remediated.

Note that it is not necessary to use the Model Base Packages feature for every OS Provisioning job. It needs only to be used once after an OS Installation Profile changes. From that point on, the Software Policy will be attached to the OS Sequence unless you remove it, and will be available for other servers as they are provisioned.

Model Base Packages Script Usage

The Model Base packages script is coded in such a way that it will function when called from another script such as `provisionOS.py` as well as being runnable as a standalone python2 pytwist script. The following are valid arguments when invoking the script:

```
model_base_packages.py --opsware-username you [--opsware-password yourpass]
--server <serverID> --ossequence <ossequenceID> [--policy_path "/Some/Folder
Path/Some Policy"]
```

Table 15 Options

Argument	Description
--version	Show the program version number and exit
-h, --help	Show this help message and exit
-u OPSWAREUSERNAME, --opsware-username=OPSWAREUSERNAME	Login username for SA
-p OPSWAREPASSWORD, --opsware-password=OPSWAREPASSWORD	Login password for SA
-s SERVER, --server=SERVER	Numeric Server ID of server to model
m POLICYPATH, --policy_path=POLICYPATH	Absolute path to the software policy that will model the packages
-e OSSEQUENCE, --ossequence=OSSEQUENCE	Numeric OS Sequence ID to link to the model software policy. If you specify an OS Sequence but not a policy path, the software policy will be created in the folder that contains the OS Sequence with the OS Sequence's name plus "Base Packages".

Reprovisioning a Managed Server

You can reprovision a managed server, but this process completely removes all data on the server as well as any network configuration settings.

Certain attributes, defined in the build script for each operating system, are preserved after you reprovision the server. For more information on OS Provisioning build scripts, see [Creating Build Customization Scripts](#) on page 71.

▶ You can only reprovision a server that runs the Solaris or Linux operating system (but not Solaris x86). Reprovisioning a VMware ESX server is not supported. Reprovisioning a server that was booted into a non-DHCP environment is not supported.

▶ For Linux reprovisioning, you can use the custom attribute `boot_kernel` to determine which kernel to which reprovision will boot. For more information, see [Custom Attributes for Linux or VMware ESX](#) on page 85.

To reprovision a managed server, perform the following steps:

- 1 From the Navigation pane, select **Devices** ▶ **All Managed Servers**.
- 2 Select a managed server to reprovision and from the **Actions** menu, select **Run OS Sequence**.

- 3 You will be shown a warning message that you are about to reprovision a managed server. By doing so, you will lose all data on the server. Click **Yes** to proceed.
- 4 In the Run OS Sequence window, please select the appropriate option before you begin the provisioning:

Yes, I understand the OS installation process will erase all data on the selected servers. (Mandatory. You must select this option in order to proceed.)
- 5 Click **Next**. In the Run OS Sequence window, select an unprovisioned server or servers to provision. To add a server, click **Add**.
- 6 Click **Next**. In the Select OS Sequence pane, click **Add** to add an OS Sequence.
- 7 Click **Next**, and in the Scheduling pane, choose if you want to run the OS Sequence, immediately, or at a later date and time.
- 8 Click **Next** and in the Notifications pane, select an email notifier. Click **Add Notifier** and enter an email address.
- 9 (Optional) Specify if you want the email to be sent upon the success of the OS Sequence job or failure of the OS Sequence job.
- 10 You can also specify a Ticket Tracking ID in the Ticket ID field.
- 11 Click **Next**, and review the OS Sequence information before you run the job.
- 12 Click **Start Job** to run the OS Sequence. When the OS Sequence has run, click **View Results** to view the results of the OS Sequence job.
- 13 When the OS Sequence job has been run, you can check the Devices ► All Managed Servers list to see the newly reprovisioned server.
- 14 (Optional) From the drop-down lists, select the manufacturer, model, or facility of the server and click **Update**.
- 15 For Intel x86 and Sun SPARC processor-based servers, locate the MAC address and Host ID of the server that you just booted.

The Lifecycle column indicates whether the server is available for OS Provisioning or if the boot into the server pool failed.

See [Support for the OGFS Agent](#) on page 91 for more information.

To obtain more information about the server, double click the server name.

Recovering When an OS Build Agent Fails to Load During PXE Boot

When an OS Build Agent fails to load on a server, the server does not appear in the Unprovisioned Server list. You can check the server console for error messages and try to boot the server again with PXE or by using the SA Boot CD.

If all errors were successfully resolved, the initial boot occurs, the OS Build Agent is loaded on the server, the server appears in the Unprovisioned Server list, and the Lifecycle column indicates that the server is available.

If you are unable to resolve the error condition, contact your SA administrator for troubleshooting assistance.

Index

B

build customization scripts

- Linux, overview, 76
- overview, 71
- requirements
 - for Linux, 78
 - for Solaris, 74
- Solaris
 - overview, 75
 - sample, 76

build images, adding hardware support for Linux, 28

Build Manager

- OS Build Agents, locating, 17

C

creating

- OS sequence, 100

custom attributes

- Linux OS provisioning, setting for, 85
- Solaris OS provisioning, setting for, 84
- Windows OS provisioning, setting for, 87

D

deleting

- media resource locators (MRLs), 42
- OS installation profiles, 70

DHCP

- Linux servers, requirements for using, 29
- OS provisioning, usage of, 103
- servers, booting with, 14
- Solaris servers, booting with, 72
- Solaris servers, usage of, 26, 106

E

editing

- media resource locators (MRLs), 41

examples

- response file
 - for Windows NT, 53
- sample Solaris build customization script, 76

F

firewall configuration, 51

firewall configuration for OS Provisioning, 51

Firewalls, 51

H

hardware preparation, overview, 25

hardware support

- adding to Linux build images, 28

histories

- viewing, changes in OS installation profiles, 69

I

installing

- OS Build Agents
 - verification, 118

L

Linux

- build customization scripts
 - overview, 76
 - requirements for, 78
- hardware support, adding to build images, 28
- PXE, using for booting servers, 26
- setting, custom attributes for servers, 85

M

media resource locators (MRLs)

- creating, prerequisites for, 34
- deleting, 42
- editing, 41

O

operating systems

- defining for OS provisioning, 59, 60
- provisioning, 103

OS Build Agents

- Build Manager, locating, 17
- failure to install, recovering from, 152
- verifying installation, 118

- OS build process
 - default values for, 83
 - OS installation profiles
 - deleting, 70
 - histories, viewing, 69
 - modifying, 66
 - modifying packages in, 68
 - overview, 49
 - properties, changing, 65
 - software, specifying, 51
 - working with, 58
 - OS media
 - prerequisites for creating MRLs, 34
 - OS Provisioning, 51
 - OS provisioning
 - hardware preparation, 25
 - Linux
 - custom attributes, setting up, 85
 - modifying operating system installation, 66
 - OS installation profiles, preparing, 59, 60
 - Prepare Operating System Wizard, 59, 60
 - SA Client
 - creating an OS sequence, 100
 - overview, 125
 - reprovisioning a managed server, 151
 - select unprovisioned servers, 125
 - Solaris custom attributes, setting up, 84
 - Solaris servers, 22
 - Windows custom attributes, setting up, 87
 - Windows servers, 23
 - OS sequence
 - attach device group, 100
 - creating, 100
 - set remediate policy, 100
- P**
- packages
 - modifying in OS installation profiles, 68
 - Prepare Operating System Wizard, 59, 60
 - prerequisites
 - MRLs, creating, 34
 - properties, OS installation profiles, changing for, 65
 - PXE images
 - overview for Windows and Linux, 26
- R**
- Red Hat Linux, 51
 - reprovisioning a managed server, SA Client, 151
 - response files
 - example
 - for Windows NT, 53
- S**
- SA Client
 - OS installation with, 125
 - scripts
 - Linux build customization scripts, 78
 - Linux servers, customizing build, 76
 - Solaris build customization scripts, requirements for, 74
 - Solaris servers, customizing build, 75
 - servers
 - reprovisioning, SA Client, 151
 - setup for servers
 - Linux OS provisioning, 104
 - operating systems for provisioning, 59, 60
 - Solaris OS provisioning, 103
 - software
 - specifying in OS installation profiles, 51
 - Solaris
 - build customization scripts
 - overview, 75
 - sample, 76
 - custom attributes, setting for Solaris servers, 84
 - OS provisioning, 22
 - requirements for build customization scripts, 74
- T**
- troubleshooting
 - OS Build Agents
 - installation failure, 152
 - verifying installation, 118
- V**
- verifying
 - installation of OS Build Agents, 118
 - viewing
 - changes for OS installation profiles, 69
- W**
- Windows servers
 - OS provisioning, 23
 - PXE, using for booting, 26
 - sample response file
 - for Windows NT, 53
 - setting custom attributes for, 87
 - wizards
 - Prepare Operating System, 59, 60