

# HPE Server Automation

Software Version: 10.23

## User Guide: Provisioning

Document Release Date: June 2016  
Software Release Date: June 2016



## 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 Notice

© Copyright 2001 - 2016 Hewlett-Packard Development Company, L.P.

### Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

## 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://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: <http://h20229.www2.hp.com/passport-registration.html>

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

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

## Support

Visit the HP Software Support Online web site at: <http://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](http://h20230.www2.hp.com/new_access_levels.jsp)

**HP Software Solutions Now** accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is <http://h20230.www2.hp.com/sc/solutions/index.jsp>

# Contents

---

<b>Chapter 1 SA Provisioning Concepts</b> .....	<b>13</b>
SA Provisioning Features .....	13
SA Provisioning Basics .....	13
The Build Plan Framework .....	14
The Service OS .....	14
Out-of-the-Box Content .....	15
<b>Chapter 2 Performing SA Provisioning</b> .....	<b>16</b>
The SA Provisioning Process .....	16
Phase 1: Preparing the Media .....	16
General Guidelines for Preparing the OS Media .....	16
HTTP/HTTPS .....	17
NFS .....	17
SMB/CIFS .....	18
Solaris 11 IPS .....	18
Phase 2: Preparing Target Servers .....	19
Target Server Requirements .....	19
Choosing How to Boot Servers .....	19
When Do I Add Servers Using iLO Registration? .....	19
When Should I Network Boot Using iLO? .....	20
When Do I Use Intelligent Provisioning? .....	20
Network Booting .....	20
IPv6 Notes .....	25
Managed Boot Clients .....	25
SA Provisioning-Supplied CD Boot Images .....	30
Embedded OS Booting (Intelligent Provisioning) .....	32
iLO Support .....	32
iLO Auto-discovery .....	32

---

Manual Registration .....	32
Customizing a Target Server for Build Plans .....	35
Using Custom Attributes .....	35
Using Device Groups .....	36
Phase 3: Running a Build Plan .....	37
Opening the Run Build Plan Wizard from Unprovisioned Servers .....	37
Opening the Run Build Plan Wizard from Managed Servers .....	38
Opening the Run Build Plan Wizard from the SA Client Library .....	39
Opening the Run Build Plan Wizard from an Open Build Plan .....	40
Starting the Build Plan .....	41
Searching for Active/Completed/Failed Run Build Plan Jobs .....	43
Personalize Network Settings .....	43
Mandatory and Optional Fields .....	44
Description of Individual Fields .....	45
Where is hpsa_netconfig Used? .....	46
Service OS with Personalized Network Settings .....	46
During the Provisioning Process .....	47
Network Personalization of an Installed System .....	47
Red Hat Enterprise Linux, CentOS, Oracle Enterprise Linux Platform .....	48
Example 1 .....	49
Extending Windows Hardware Support .....	51
Joining a Windows Domain or Workgroup .....	52
Provisioning an Already Provisioned Server .....	52
Automatic Reprovisioning .....	52
Manual Reprovisioning .....	53
Device Naming .....	53
<b>Chapter 3 SA Provisioning Common Use Cases .....</b>	<b>56</b>
Provisioning Windows-Based Servers .....	56
Provisioning Linux-Based Servers .....	63
Provisioning Solaris x86-Based Servers .....	69
Provisioning ESXi-Based Servers .....	75

---

Updating an HP ProLiant Server's Firmware .....	83
Downloading an SPP .....	83
Deploying an SPP Image to Your Media Server .....	83
Preparing and Running the "ProLiant SW - Offline Firmware Update" Build Plan .....	84
Additional Parameters for the "Update Firmware Using SPP" Build Plan Step .....	85
<b>Chapter 4 Creating New SA Build Plans .....</b>	<b>88</b>
Customizing Out-of-the-Box Build Plans .....	88
Editing a Build Plan .....	89
Custom Attribute Substitution .....	90
Customizing Installation Profiles .....	91
Modifying an Existing Installation Profile .....	91
Network Setup .....	95
Firewall Considerations .....	95
Flow Control Mechanism .....	95
Build Plan Steps .....	97
Run Script Step .....	97
Install Zip Step .....	98
Capture and Deploy Configuration File Steps .....	98
Add to Device Group Step .....	98
Attach Software or Patch Policy Steps .....	98
Remediate Server Step .....	99
Managing a Server's State .....	99
Asserting the State of the Server .....	99
Changing the State of a Server .....	99
Using Scripts as Building Blocks .....	99
Running Build Plans on a Managed Server .....	100
Should I Use Server Scripts or OGFS Scripts? .....	100
<b>Chapter 5 SA Provisioning Administration .....</b>	<b>102</b>
Required Permissions .....	102
SA Provisioning Components .....	102
DHCP Configuration (IPv4 and IPv6) for SA Provisioning .....	102

---

DHCP Software Included with the SA Provisioning Boot Server .....	102
SA DHCP Server (dhcpd) .....	103
SA dhcpd.conf File .....	103
SA dhcpd6.conf File .....	103
SA DHCP Network Configuration Tools (dhcpdtool and dhcpd6tool) .....	104
Required Information for the SA DHCP Network Configuration Tool for IPv4 (dhcp- dtool) .....	104
Required Information for the SA DHCP Network Configuration Tool for IPv6 (dhcp- d6tool) .....	104
Configuring the SA DHCP IPv4 Server for SA Provisioning .....	105
Configuring the SA DHCP IPv6 Server for SA Provisioning .....	107
Starting and Stopping the SA DHCP Server for IPv4 and IPv6 .....	109
Modifying the dhcp.conf File for Use with WINPE .....	109
Configuring an Existing ISC DHCP Server for SA Provisioning .....	110
Configuring a Windows DHCP Server for SA Provisioning .....	110
Controlling the SA and Windows DHCP Servers' Responses to SA Provisioning Requests .....	112
Enabling IBM POWER6 SA Provisioning with the DHCPD Tool .....	113
<b>Chapter 6 OS Sequence-Based Provisioning Requirements, Setup, and Usage .....</b>	<b>116</b>
The OS Sequence Provisioning Process .....	116
SA OS Provisioning Components .....	118
The OS Build Agent .....	118
The Build Manager .....	118
The Media Server .....	119
The Boot Server .....	119
Build Customization Scripts .....	119
How the OS Build Agent Locates the Build Manager .....	119
WinPE .....	119
Linux: .....	120
Linux IA64: .....	120
Oracle Solaris/Sun SPARC 10 and 11 .....	120
Non-DHCP Environments .....	120

---

Provisioning Setup for OS Sequences .....	120
OS Provisioning Setup Task Summary .....	121
Setting Up the Media Server .....	122
Creating Media Resource Locators (MRLs) .....	123
Import Media Tool Prerequisites .....	123
Import Media Tool Syntax and Options .....	123
Configuring the Media Server for Microsoft Windows OS Media/Image .....	125
Importing Windows Media from Linux Host .....	125
Importing Windows Media from a Solaris Host .....	126
Configuring the Media Server for Windows Server 2003 (x86/x86_64), 2008, 2008 R2 x64, and 2012 OS Media .....	126
Windows Media: Preparing Network Driver Directories .....	126
Windows Media: Hosting Windows Media on a Windows 2K Server Using a Share ..	127
Configuring the Media Server for Red Hat Linux or VMware ESXi OS Media .....	127
Configuring the Media Server for SUSE Linux or SUSE Enterprise Linux OS Media ..	128
Configuring the Media Server for Oracle Sun Solaris 10 .....	130
Configuring the Media Server for Oracle Sun Solaris 11 .....	131
Oracle Solaris Automated Installer .....	131
Enabling Oracle Solaris 11 x86 with the Manage Boot Client .....	132
Steps to Create MRLs .....	132
Media Resource Locator Administration .....	133
Advanced Import Media Tool Information .....	134
Multipath SAN Support for OS Provisioning .....	134
OS Sequences .....	135
SUSE Linux Enterprise Server 11 .....	135
Configuring RAID on HP ProLiant Servers Before OS Provisioning .....	137
Supported Hardware .....	137
Supported Operating Systems .....	137
Capture a Baseline HP ProLiant RAID Configuration .....	138
Creating an HP ProLiant RAID Dynamic Server Group .....	141
Manually Specifying an HP ProLiant RAID Configuration .....	141
Defining Installation Profiles and OS Sequences .....	141



---

OS Installation Profile Requirements .....	141
Overview .....	141
Specifying Software for OS Provisioning .....	143
Configuration Files .....	143
Oracle Solaris/Sun SPARC 10 Installation Profile Requirements .....	144
Red Hat Linux Installation Profile Requirements .....	144
VMware ESX Installation Profile Requirements .....	145
SUSE Linux Installation Profile Requirements .....	145
Microsoft Windows Installation Profile Requirements .....	145
Sample Response File for Windows Server 2003 .....	146
Defining and Managing OS Installation Profiles .....	147
Install Profile Tool Syntax and Options .....	147
Defining an OS Installation Profile — Linux/UNIX .....	151
Defining an OS Installation Profile — Windows .....	151
Hardware Signature Files for Windows .....	153
Modifying Existing OS Installation Profiles .....	153
Changing the OS Installation Profile Properties .....	154
Modifying the OS Installation Profile Packages .....	154
Viewing Change History for an OS Installation Profile .....	154
Deleting an OS Installation Profile .....	155
Creating Build Customization Scripts .....	155
Using Build Customization Scripts .....	155
Solaris Build Customization Scripts .....	156
The Sun Solaris Build Process .....	156
Requirements for Solaris Build Customization Scripts .....	158
Solaris Provisioning from a Boot Server on a Red Hat/SLES 10 Linux Server .....	159
Creating a Solaris Build Customization Script .....	160
Sample Solaris Build Customization Script .....	160
Linux Build Customization Scripts .....	161
Linux/Itanium Build Process .....	162
Sample Solaris Build Customization Script .....	163

---

Requirements for Linux Build Customization Scripts .....	164
VMware ESX Build Process .....	165
VMware ESX Build Customization Scripts .....	165
Windows Build Customization Scripts .....	165
Windows Build Process (WinPE Boot Image) .....	165
Legacy Build Customization Script run.bat .....	167
Creating a Windows Build Customization Script (WinPE) .....	167
Sample run.cmd File .....	168
Defining Custom Attributes .....	170
Custom Attributes for Sun Solaris 10 and 11 .....	170
Custom Attributes for Linux or VMware ESX .....	172
Using the boot_disk Custom Attribute to Specify the Boot Drive .....	174
Custom Attributes for Microsoft Windows .....	175
Adding Custom Attributes to OS Installation Profile (SA Client) .....	177
Creating OS Sequences .....	178
OS Sequence Contents .....	178
Defining an OS Sequence .....	179
The Manage Boot Clients Option .....	181
Requirements .....	182
Required Permissions .....	182
Installation .....	182
Using the Manage Boot Clients Option .....	182
Running an MBC APX .....	183
The MBC Form-Based Method (Web-Based) .....	183
The MBC APX Command-Line Interface .....	183
Special Attributes for the CLI and CSV Input Form .....	184
CSV Input Files .....	185
Special Attributes for DHCP Reconfiguration .....	187
iLO Integration .....	187
Bootting a Red Hat Enterprise Linux Server in a Non-DHCP Environment .....	188
DHCP Custom Attribute .....	189

---

Booting a Red Hat Enterprise Linux Itanium 64-bit Server in a Non-DHCP Environment Using Elilo Boot .....	190
DHCP Custom Attribute .....	192
Booting a Windows Server in a Non-DHCP Environment .....	192
Booting an Unmanaged Windows Server in a Non-DHCP Environment .....	193
DHCP Custom Attribute .....	194
<b>Chapter 7 HP-UX Provisioning .....</b>	<b>196</b>
Prerequisites .....	196
Ignite Setup on the SA Core .....	196
APXs .....	197
Customer Configuration Subfolders .....	197
Permissions .....	198
Installing an Operating System on HP-UX Servers .....	199
Creating a Custom Configuration .....	199
Boot Target .....	204
Provision the Target Servers .....	205
Deleting Custom Configurations .....	212
Glossary .....	213
Useful Links .....	214
Troubleshooting .....	214



# SA Provisioning Concepts

HP Server Automation (SA) provides the ability to provision servers out-of-the-box, including, but not limited to, the base operating system for a variety of targets both physical and virtual.

You can also use SA to provision firmware and applications and any other steps required to promote servers into production.

SA can also reliably and consistently provision a large number of operating systems with no manual intervention. SA includes both out-of-the-box content and an extensive framework that allows you to customize the way servers are provisioned.

## SA Provisioning Features

SA Provisioning has the following benefits:

- **Provisioning that simply works right out of the box**

SA Provisioning provides you with a distinct advantage over other operating system installation and multi-purpose provisioning tools by providing a reliable, intuitive interface that is consistent no matter the configurations being provisioned. SA provides a set of baseline Build Plans (provisioning templates) that, out of the box, can be used to provision and configure almost all SA supported operating systems. You can easily customize your own Build Plans by copying an existing plan and modifying it for your specific needs.

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

SA Provisioning supports many types of servers, networks, security architectures, and operational processes across a large variety of hardware models. This flexibility ensures that you can provision operating systems to suit your organization's needs.

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

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

- **Integration with other SA features**

Because SA 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 essential to high-quality operations and delivery of reliable change management.

## SA Provisioning Basics

SA Provisioning has three essential parts:

- The Build Plan framework
- A minimal service OS
- Out-of-the-box content

## The Build Plan Framework

A *Build Plan* consists primarily of a list of steps that can be executed against a *target server* (the server to be provisioned). These steps specify the various tasks that ultimately provision a server. The framework includes an execution engine that runs the steps sequentially. This makes understanding Build Plans and their actions easy and intuitive.

Build Plans are SA objects, and as such, they can be viewed and manipulated in the SA Client Library.

A number of baseline Build Plans are included when an SA Core is installed. These Build Plans can, by default, perform many common provisioning tasks.

Multiple Build Plan steps types are supported such as running a script, deploying a zip package or deploying a configuration file with parameter substitution.

Other step types integrate with various SA features and provide the ability to attach software policies, start remediation, join a device group and more.

## The Service OS

SA ships with several minimal RAM-based operating systems that can be started over a network, from physical or from virtual media (CD, DVD or ISO images), on a physical server or virtual machine.

These are called *Service OSes*.

Because of the limited functionality within a service OS, only a subset of SA operations can be run against a server that has been booted into a service OS. The primary purpose of a service OS is to enable installation of a full operating system, also known as a *Production OS* as well as other maintenance tasks that cannot be performed from a production OS.

The service OS is the means by which the Build Plan framework gathers information and executes tasks on a target server.

These service OSes make use of a special instance of the SA Agent configured to run in this limited environment. Starting a server in the service OS is also known as *bringing the server into Maintenance mode*. Servers in Maintenance mode are recognizable by the icon that is displayed in the SA Client and by their **Maintenance** status.

### Maintenance Icon



Service OSes can be booted on any server, whether or not it has an installed OS. A server can safely be booted from a production OS into Maintenance mode and back.

**Note:** While running the Service OS is not in itself destructive, you can execute destructive actions inside the Service OS, like erasing the disk so care must be taken.

## Out-of-the-Box Content

The out-of-the-box content is a collection of Build Plans populated with Build Plan steps that deliver functionality for common use-cases and are the basic building blocks for user-created Build Plans.

This includes provisioning of operating systems, network configuration and end-to-end provisioning of HP ProLiant servers (including firmware and hardware configuration management).

Using this functionality is as simple as running the Build Plans.

The out-of-the-box content is installed as part of SA Core installation or upgrade.

Installation-specific parameters can be customized either by editing the command-line arguments of script steps in the Build Plans or by specifying *Custom Attributes*. These are a generic parameter passing mechanism in SA. The custom attributes that influence the behavior of an out of the box Build Plan are set with blank values on the Build Plan object.

Custom attributes can be specified using a Build Plan or can be specified at the server, device group or facility level. See [Defining Custom Attributes](#).

SA Provisioning uses a *Media Server* to serve large objects like OS installation media, system images (for example, Windows WIM images) and collections of drivers and firmware to the server being provisioned.

Multiple transfer protocols are supported: HTTP, HTTPS, NFS and SMB (Windows shares).

See the *SA Provisioning Matrix* to determine which protocols are supported with the operating system(s) you want to provision.

SA includes a media server that can serve media through SMB and NFS. The Media Server is installed when you choose to install the SA Provisioning Components.

**Note:** You can use any server that supports the transfer protocols above to store and serve media. You will specify the location of the media server in the Build Plan.

# Performing SA Provisioning

This section describes the SA Provisioning process.

## The SA Provisioning Process

This section will guide you through provisioning servers in three easy steps using only the out-of-the-box provisioning content (baseline Build Plans).

The Provisioning process consists of several phases:

[Phase 1: Preparing the Media](#)

[Phase 2: Preparing Target Servers](#)

[Phase 3: Running a Build Plan](#)

If you want to customize how SA performs provisioning, see [Creating New SA Build Plans](#).

You can find examples in [SA Provisioning Common Use Cases](#).

**Note:** Ensure that the DHCP Server has been configured for SA Provisioning as described in [DHCP Configuration \(IPv4 and IPv6\) for SA Provisioning](#).

See [Creating New SA Build Plans](#) for advanced information and [SA Provisioning Common Use Cases](#) for sample step-by-step procedures.

## Phase 1: Preparing the Media

Depending on platform and operating system, one or more of the following protocols are supported for provisioning using Build Plans:

- HTTP/HTTPS
- NFS
- SMB/CIFS

See the Provisioning section of the *SA Support and Compatibility Matrix* for more information about supported platforms and protocols.

In addition, Solaris 11 installations will require an IPS package server, provided either by Oracle Corp. or hosted in-house.

### General Guidelines for Preparing the OS Media

- Unpack the vendor provided ISO(s) on the media server in a shared path.

For example, extracting an ISO on a Linux system:

```
mkdir -p /mnt/media
```



```
mount -o loop,ro /path/to/media.iso /mnt/media
cp -ar /mnt/media /media/opsware/os/version
chown -r nobody:nobody /shared/path/extracted_media
umount /mnt/media
```

- Ensure that the Media Server does not alter file paths. For this reason HP recommends that you use a Linux-based Media Server.
- See the **Set Media Source** step in the Build Plan you want to run. Double-check the protocol, media server and share path to the media. Note that some deployments support multiple protocols.
- Verify that the processor architecture of the media matches the architecture of the Build Plan you are using.

### HTTP/HTTPS

OS installations that support fetching media through HTTP/HTTPS provide flexibility when accepting media sources. You can use an in-house server, an external server or an official mirror (this allows you to bootstrap a media server without handling OS installation yourself).

See the documentation for your preferred HTTP/HTTPS serving solution for instructions about how to set up a share.

- Apache HTTPD documentation:

<http://httpd.apache.org/docs/>

- Nginx documentation:

<http://nginx.org/en/docs/>

- Microsoft IIS documentation:

<http://www.iis.net/learn>

**Note:** Accessing a media server using HTTP/HTTPS through a proxy is *not* supported.

### NFS

OS Media files shared through NFS must be read-accessible by any user from the servers that require provisioning. Check and adjust file and directory permissions as necessary.

See the documentation for your operating system for information about how to export NFS shares.

The following is a sample `/etc/exports` file, which exports the directory `/media` (and all its subdirectories) as read-only to all hosts:

```
/media *(ro)
```

Run the following command to test-mount the NFS share from a client machine:

*Linux*

```
mount -F nfs -o ro <media-server-host>:/media /mnt/
```

#### **Solaris**

```
mount -F nfs -o ro <media-server-host>:/media /mnt/
```

The SA Core has, by default, two NFS exports dedicated to sharing OS Media activated.

These exports are listed in `/etc/exports` as:

```
/media/opsware/linux *(ro,no_root_squash,async,insecure)
/media/opsware/sunos *(ro,no_root_squash,async,insecure)
```

#### **SMB/CIFS**

SA Provisioning supports SMB shares hosted on a variety of platforms, including Windows, Linux or UNIX servers and dedicated NAS appliances. Files can be accessed either through guest accounts or secured with username/password.

For more information about configuring SMB shares under Windows, see the Windows documentation at:

<http://technet.microsoft.com/en-us/library/cc770406.aspx>

For more information about configuring SMB shares under UNIX platforms using Samba, see the Samba documentation at:

<https://www.samba.org/samba/docs/>

The SA Core provides an SMB share named `OSMEDIA`, which is active by default, in:

```
/media/opsware/windows
```

#### **Solaris 11 IPS**

In order to successfully complete provisioning, Solaris 11 Build Plans require an IPS repository that is accessible through HTTP in addition to the installation media.

**Note:** You *must* use a dedicated IPS server. Serving a local file-based repository through HTTP using a generic server (e.g. Apache, Nginx) is *not* supported.

The simplest Solaris 11 IPS solution is to use the central Solaris 11 IPS repository, available at:

<http://pkg.oracle.com/solaris/release/>

However, you may want to host an in-house repository for performance, package validation before entering production, inclusion of custom packages etc.

See the instructions for setting up and administering a local repository at:

[http://docs.oracle.com/cd/E23824\\_01/html/E21803/repo\\_int1.html](http://docs.oracle.com/cd/E23824_01/html/E21803/repo_int1.html)

**Note:** In addition to these instructions, HP recommends setting up a reverse caching proxy on top of your local IPS server. Performance testing has shown that a non-cached IPS server cannot successfully provide packages to more than five simultaneous Solaris 11 deployments.

## Phase 2: Preparing Target Servers

This section describes setting up and configuring your provisioning target servers.

### Target Server Requirements

Before you can discover servers for SA Provisioning, you must ensure that the servers are set up properly and meet the following requirements:

- Ensure that at least one network interface is configured. If you are going to network boot, the deployment interface must be on the same network as the SA Boot infrastructure. Otherwise, at least one SA Agent Gateway must be reachable.
- There can be only one network interface (NIC) attached to the deployment network.
- For HP ProLiant servers, ensure that the iLO is connected to the network and is reachable by SA through the same SA Realm as the server it controls.

**Note:** Realms are an SA construct that allow SA to manage servers on different networks in the same Facility without fear of IP address conflicts. A realm is a unique identifier, appended to the IP address of a device in a Facility's network, that allows SA Gateways to uniquely identify devices on different networks in a Multimaster Mesh that may have conflicting IP addresses. See also the *SA Overview and Architecture Guide*.

### Choosing How to Boot Servers

SA Provisioning supports the following methods to boot a server:

- **Network booting:** suitable for fully automated deployments on heterogeneous hardware and virtual machines. This requires that the server be on the same network as the SA Boot Server and that the SA network boot infrastructure is configured (DHCP and PXE server is running).
- **CD booting:** (IPv4 only) targeted for environments where network booting and DHCP are not configured or the server is not in the same network as the SA boot server.
- **Embedded OS booting:** eliminates the need to configure an SA network boot infrastructure without sacrificing automation. Only available for HP ProLiant Gen8 or newer servers and is also known as *Intelligent Provisioning*.

### When Do I Add Servers Using iLO Registration?

- You have an HP ProLiant server with iLO 2 or newer remote management available.
- You have the iLO credentials for your target server.

- You have an HP ProLiant Gen8 or newer server and you do *not* want to use DHCP and network boot.
- You do not want a special access account automatically created on your iLOs. See [iLO Support](#).

### **When Should I Network Boot Using iLO?**

- You want to discover all the server information so you can see it and use it for search in the SA Client prior to provisioning.
- You want to verify the server network connection before running a Build Plan.
- You want to see the server listed by its default DNS name.
- You do *not* have iLO credentials for your target servers.
- You do not want to use Intelligent Provisioning on your ProLiant Gen8 or newer servers.
- You prefer the simplicity of a power-on discovery because your servers automatically network boot.
- You have a large number of servers and you find it more practical than compiling a list of iLO network addresses and credentials.

### **When Do I Use Intelligent Provisioning?**

- You want to run a Build Plan immediately and do not want to manually boot the server.
- You want to leave the server powered off until you are ready to install it.
- All your servers are of the same type so you do not need the full properties information.

### **Network Booting**

SA provisioning supports network booting X86 and X86\_64 target servers. For information about IPv4 and IPv6 support, see the Support and Compatibility Matrix. UEFI network booting support is also available but only on UEFI-capable HP ProLiant servers.

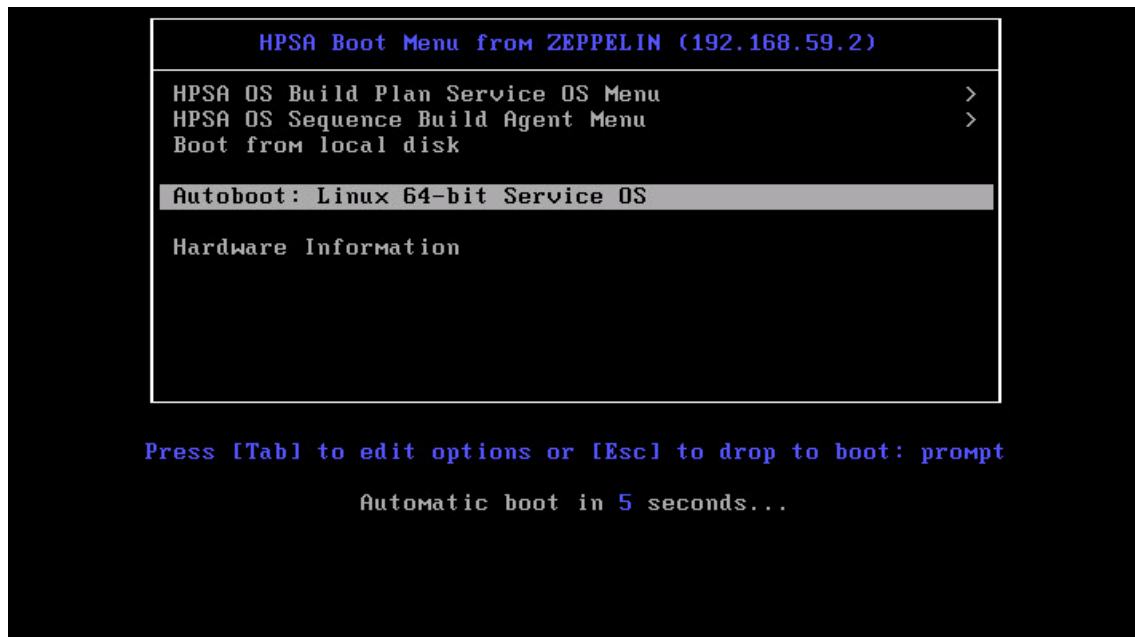
Network booting using IPv6 only is not currently supported. However, you can network-boot using IPv4 into an IPv6-only service OS, making it possible to leverage IPv6-only infrastructure, such as accessing an IPv6 media server. Note that the "Personalize Network Settings of Installed System" can also be used to enable and disable networking stacks, so IPv6 can also be enabled after the OS is installed.

SA, by default, is configured to network boot servers that are not yet registered with the SA Core into the "Linux" boot option. This enables you to bring a server into Maintenance mode simply by ensuring that the target server requirements are met and powering on the server. Subsequent boots will default to the "Local disk" boot option.

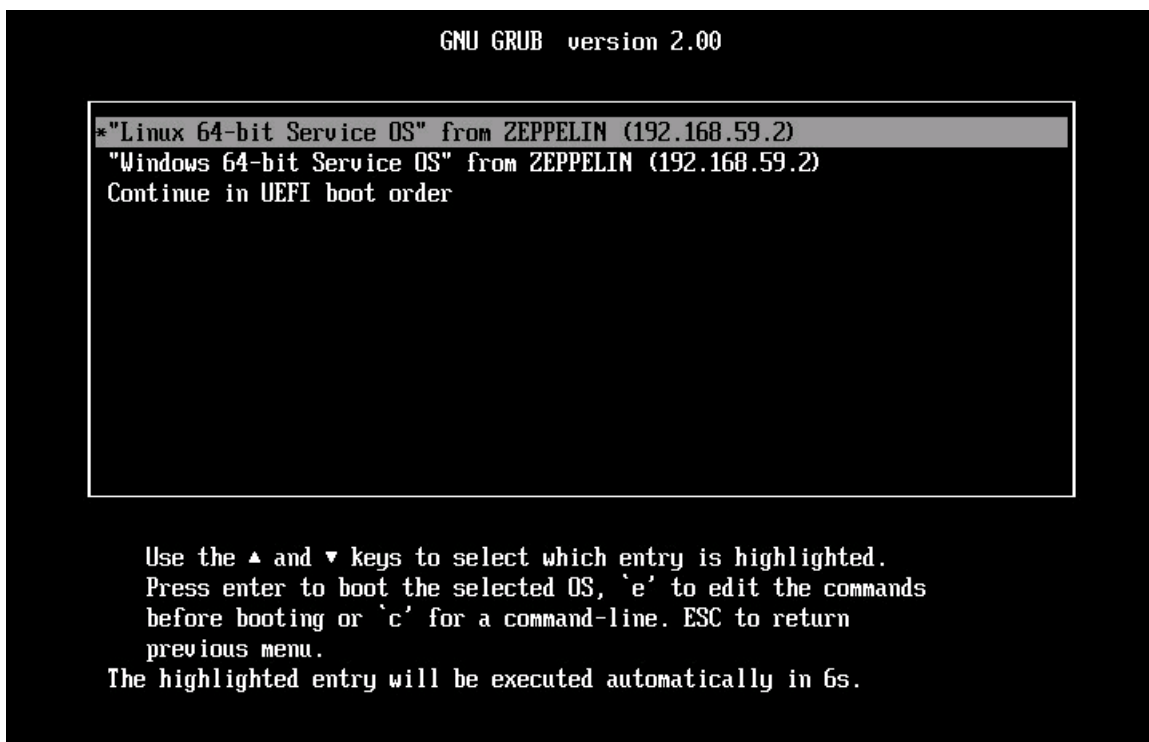
You can network-boot manually to a desired maintenance OS, but booting can also be controlled using the Managed Boot Clients Web Extension or by adding a "Boot" step to a Build Plan. If iLO support is available, selecting the desired network boot option and powering on the server is also

handled by the "Boot" step. For servers without iLO, HP recommends specifying the "Network" boot option in the boot order first, so you can boot a server to a maintenance OS without intervention.

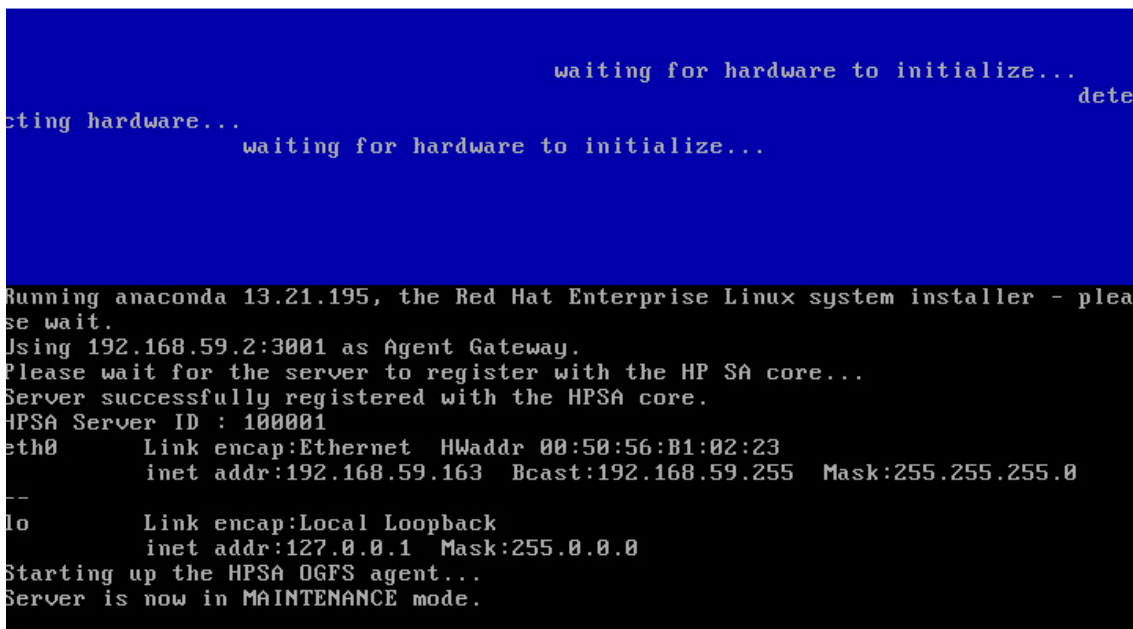
From the network boot menu for a legacy BIOS server, you can boot a 32/64-bit Linux, Windows PE, or Solaris maintenance OS.



From the network boot menu for a UEFI server, you can boot a 64-bit Linux or Windows PE maintenance OS.



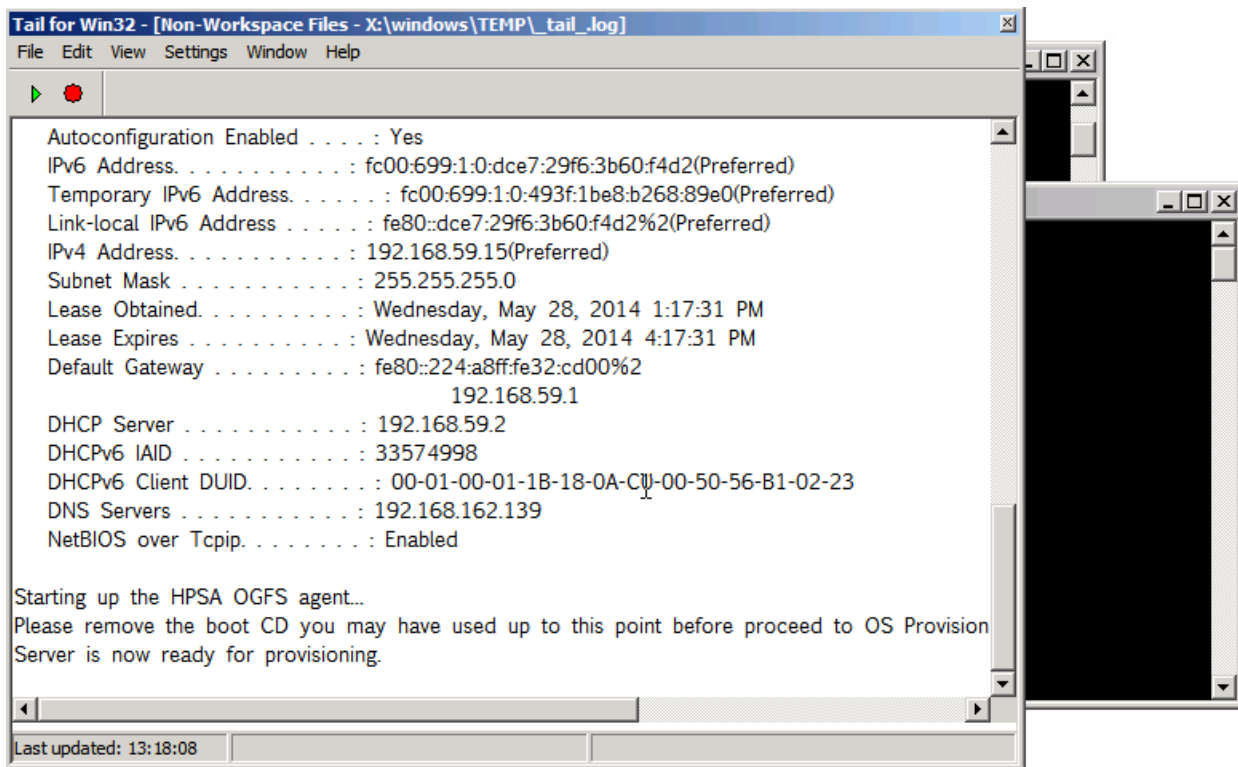
After the server is running a Maintenance mode Linux OS, you see a screen similar to the following for IPv4:



or for IPv6:

```
Running anaconda 13.21.195, the Red Hat Enterprise Linux system installer - please wait.
Using fc00:377:1::2:3001 as Agent Gateway.
Please wait for the server to register with the HP SA core...
Server successfully registered with the HPSA core.
HPSA Server ID : 2550001
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
Starting up the HPSA OGFS agent...
Server is now in MAINTENANCE mode.
```

or, in a Maintenance mode Windows PE OS (IPv4 or IPv6):

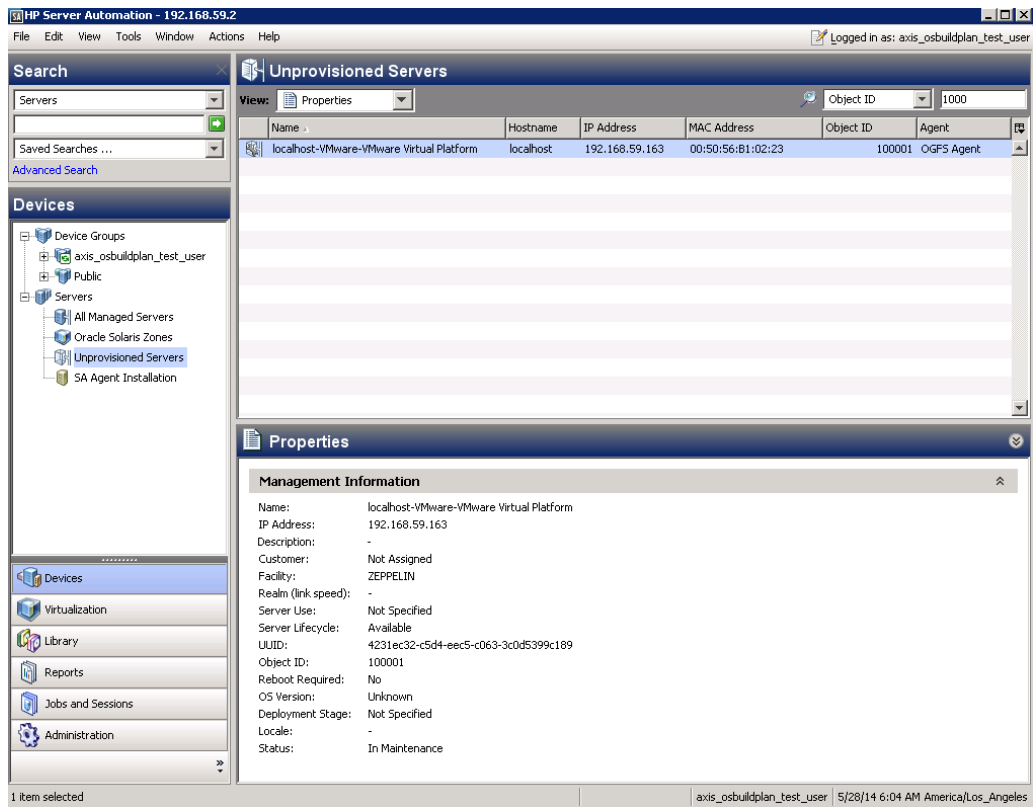


WinPE x64 (64 bit)  
HP boot image version: 55.0.50290.0



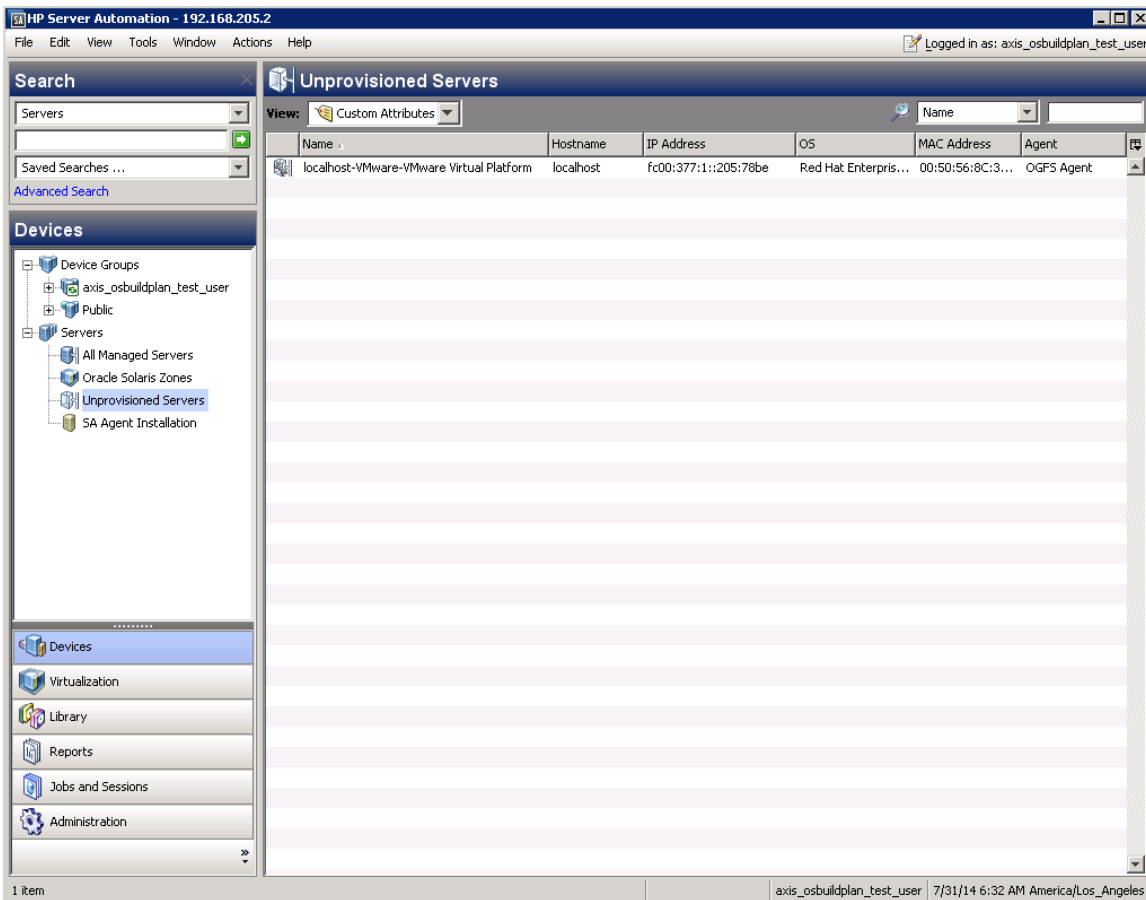
Note the identifying information, such as the SA object ID (machine ID, or MID), IP address, or MAC, so you will be able to find the device in the SA Client’s Unprovisioned Servers list.

For IPv4:



For IPv6:





## IPv6 Notes

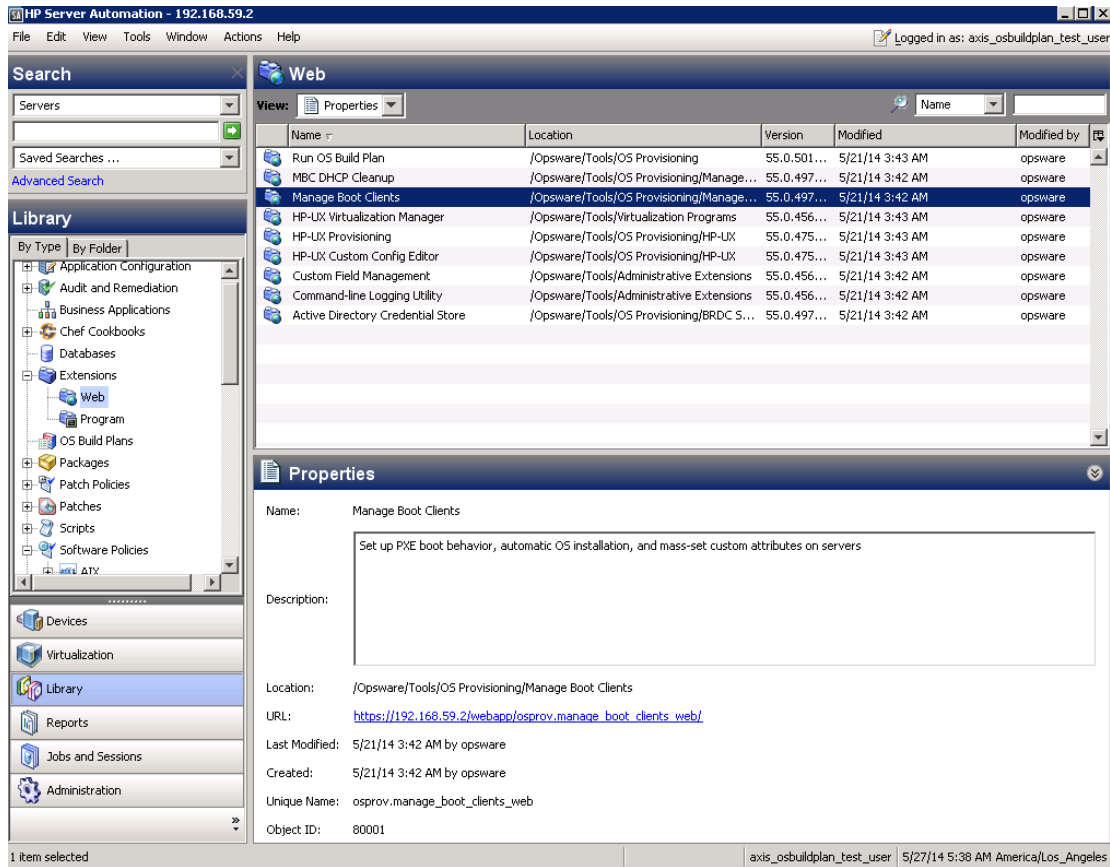
**Service OS IPv6:** Because of some limitations, in an IPv6 scenario the Service OS PXE boots using IPv4, and after that IPv6 is configured.

**Difference between IPv6 Install and IPv4 + PINP IPv6 Install:** In an IPv6 installation, the entire installation process is performed in an IPv6 environment, so you need an IPv6 media server. For IPv4 + PINP IPv6, the installation is performed in an IPv4 environment, and after installation is complete, the server is configured for IPv6.

## Managed Boot Clients

The Managed Boot Clients Web Extension allows you to create a server record with a pre-configured network boot option or configure network booting for an existing server record matched by a MAC address.


It also enables you to select a Build Plan to run automatically when the server reaches Maintenance mode. To do this, run the extension and perform the steps below.



1. Define a server.

You can also register an iLO with the target server during this step. Select “Enabled iLO Settings” for additional input.

**Managed Boot Clients - General Form**

Specify the server's unique MAC Address and optionally select which Customer and Facility the server should be assigned to. 


* MAC Address:	<input type="text" value="00-01-02-03-04-05"/>
DHCP Hostname:	<input type="text" value="showcasehostname.dev.sa.hp.com"/>
DHCP IP Address:	<input type="text" value="192.168.59.13"/>
E-mail notification on failure:	<input type="text"/>
E-mail notification on success:	<input type="text"/>
Customer:	<input type="text" value="Not Assigned"/>
Facility:	<input type="text" value="ZEPPELIN"/>

Enable iLO Settings

Multiple Client Form... Next >

## 2. Select a boot image and a Build Plan

**Managed Boot Clients - OS-Specific Parameters**


Configure how the server will reach the unprovisioned server pool. You may choose between an OS Build Plan or an OS Sequence to be automatically run, eliminating the need to use the Run OS Build Plan or the Install OS Sequences wizards. 

* Configure automatic provisioning using:	<input checked="" type="radio"/> OS Build Plans <input type="radio"/> OS Sequences
* PXE image:	<input type="text" value="winpe64_40-ogfs"/>
OS Build Plan:	<input type="text" value="Windows 2012 R2 x64 Default Install"/>

< Back Next >

3. Customize by adding custom attributes, if required, and press **Start**.

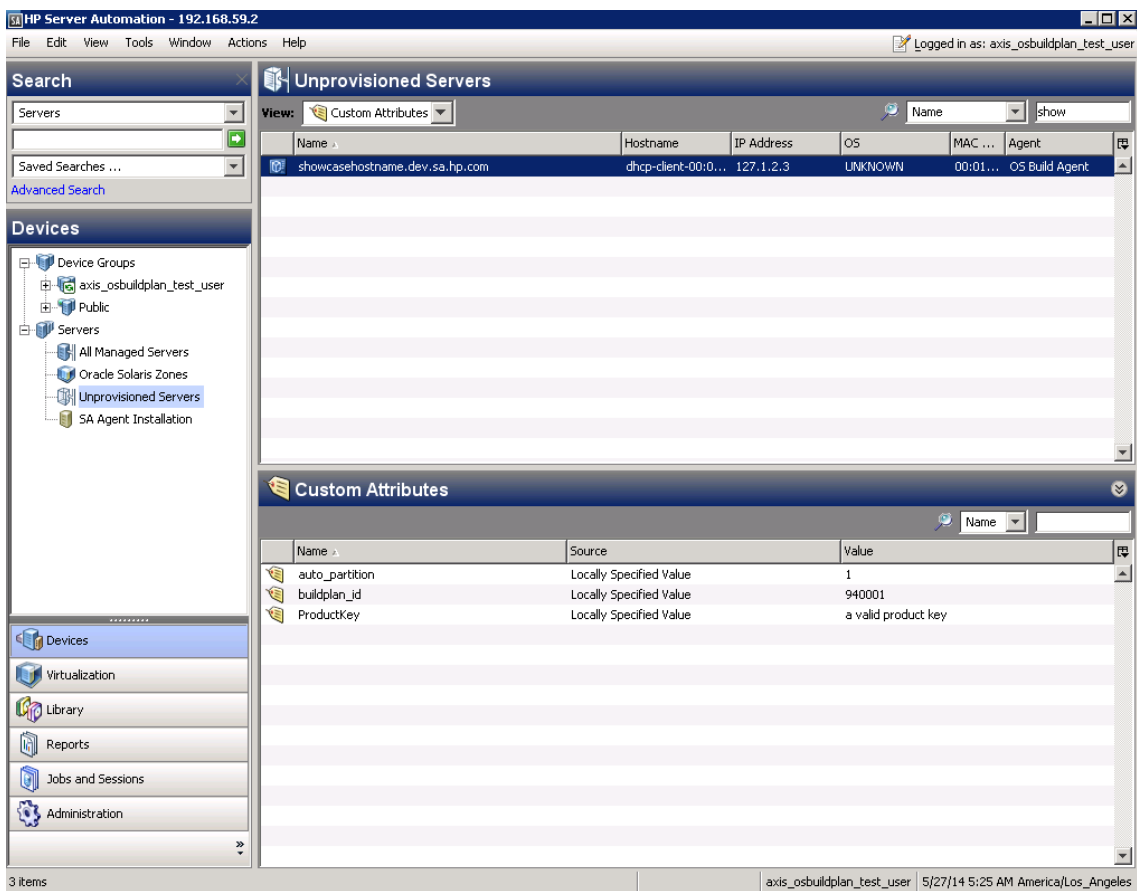
**Managed Boot Clients - Add Custom Attributes**

Custom attribute name/value pairs may be associated with the server record by adding them to the table below. 

Name	Value
<input type="text" value="ProductKey"/>	<input type="text" value="a valid product key"/>
<input type="text"/>	<input type="text"/>

Ticket ID:

After the Build Plan job completes, a Pre-Unprovisioned server record is created with any defined Custom Attributes.



**Note:** The Managed Boot Clients Web Extension configures DHCP only on request. It does *not* support concurrent configuration for one or multiple devices. When configuring multiple devices, the same Managed Boot Clients process must be used.

For the example above, the following CSV was used:

```
`00:13:E8:9A:93:AA,pxe_image-
e=winpe32,dhcp.ip=10.2.3.10,dhcp.hostname=m0010,customer=Opsware,dns_
server=10.6.4.2,root_password=wealth`
```

```
`00:13:E8:9A:93:AB,pxe_image-
e=winpe32,dhcp.ip=10.2.3.11,dhcp.hostname=m0011,customer=Opsware,dns_
server=10.6.4.2,root_password=wealth`
```

```
`00:13:E8:9A:93:AC,pxe_image-
e=winpe32,dhcp.ip=10.2.3.12,dhcp.hostname=m0012,customer=Opsware,dns_
server=10.6.4.2,root_password=wealth`
```

```
`00:13:E8:9A:93:AD,pxe_image-
e=winpe32,dhcp.ip=10.2.3.13,dhcp.hostname=m0013,customer=Opsware,dns_
server=10.6.4.2,root_password=wealth`
```

The Managed Boot Clients Web Extension returned:



Additionally, DHCP is configured in `/etc/opt/opsware/dhcpd/dhcpd_abc.conf`:

```
# Begin Opsware added hosts (do not edit)
host showcasehostnamedevsahpcom {
    hardware ethernet 00:01:03:04:05:06;
    option host-name showcasehostname;
    fixed-address 192.168.59.13;
}
# End Opsware added hosts (do not edit)
```

## SA Provisioning-Supplied CD Boot Images

SA Provisioning provides several service operating system boot CD images (ISOs) that you can burn to CD/DVD. These images enable you to bring a server into maintenance without having configured DHCP.

You can also configure these ISO images in virtual machine CD-ROM drives or mount them using iLO Virtual Media or similar technology.

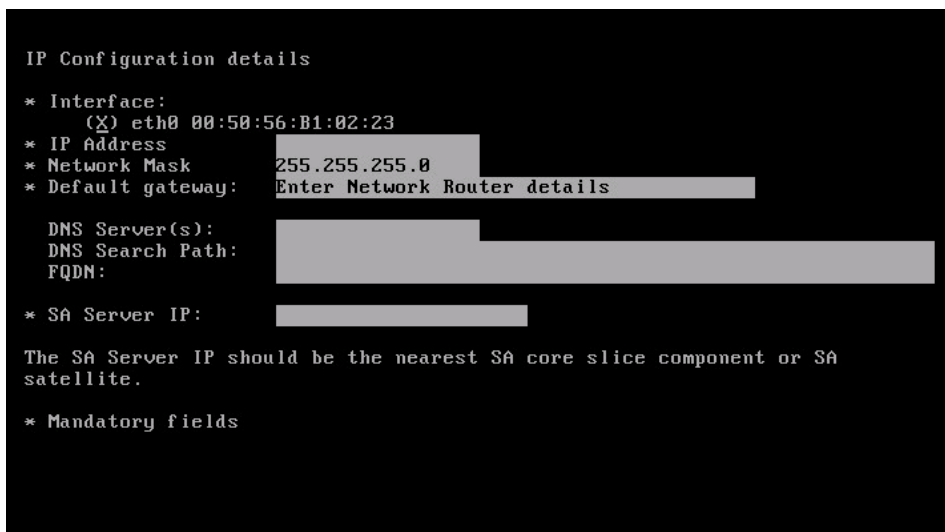
To download the desired ISO, open the SA Client and in the Navigation Pane, go to the Library tab then select the “By Folder” view and navigate to:

```
/Opsware/Tools/OS Provisioning/
/Opsware/Tools/OS Provisioning/WinPE
```

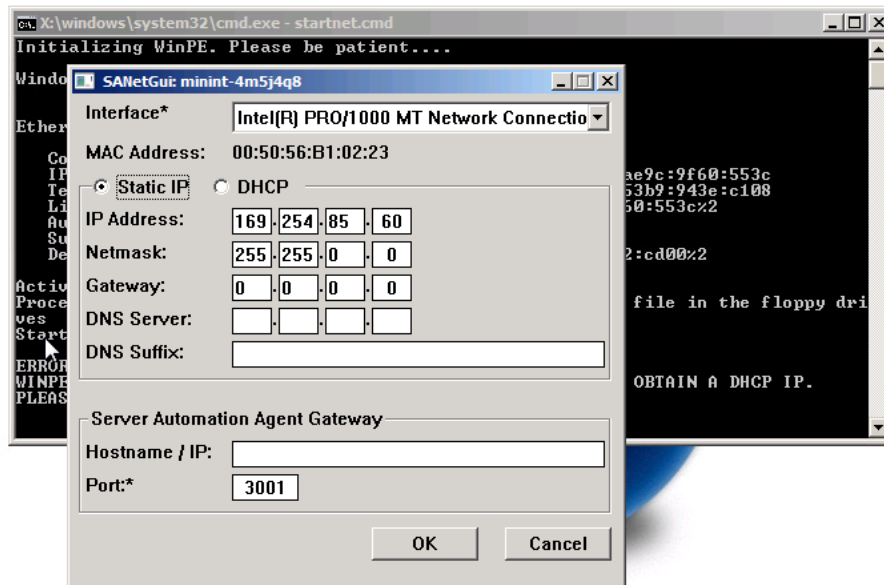
Select the desired ISO image and from the **Actions** menu select **Export software...**

After booting the CD, you can use the boot configuration screen to enter all the information needed to bring a server into Maintenance mode. The settings that you specify here are preserved on the server in the `hpsa_netconfig` custom attribute.

Maintenance mode Linux OS static boot configuration screen:



Maintenance mode Windows PE OS static boot configuration screen:



WinPE x64 (64 bit)  
HP boot image version: 55.0.50290.0

## Embedded OS Booting (Intelligent Provisioning)

Embedded OS booting is supported *only* for HP ProLiant Gen8 or newer models. This boot method's advantage is that you can move a server into Maintenance mode without enabling network booting or configuring DHCP as long as you provide static IP information.

Note that embedded OS booting must be initiated from the SA-provided "Boot" step and you can boot either a 64-bit maintenance mode Windows PE or Linux OS. See the "Boot" step description for more details.

## iLO Support

iLO support is provided by SA to enable operations like:

- Power control
- Querying or changing the one-time boot option
- Querying or changing the server boot mode (Legacy or UEFI) for UEFI capable HP ProLiant
- Instructing HP ProLiant Gen8 or newer model servers to boot from the embedded OS (Intelligent Provisioning)

iLO support is enabled in SA, either automatically (see [iLO Auto-discovery](#)) or through manual registration when an iLO manager is associated with a server.

### iLO Auto-discovery

When bringing an HP ProLiant server with iLO 2 or newer into a Maintenance mode OS, SA automatically starts an iLO registration job to associate an iLO manager with that server.

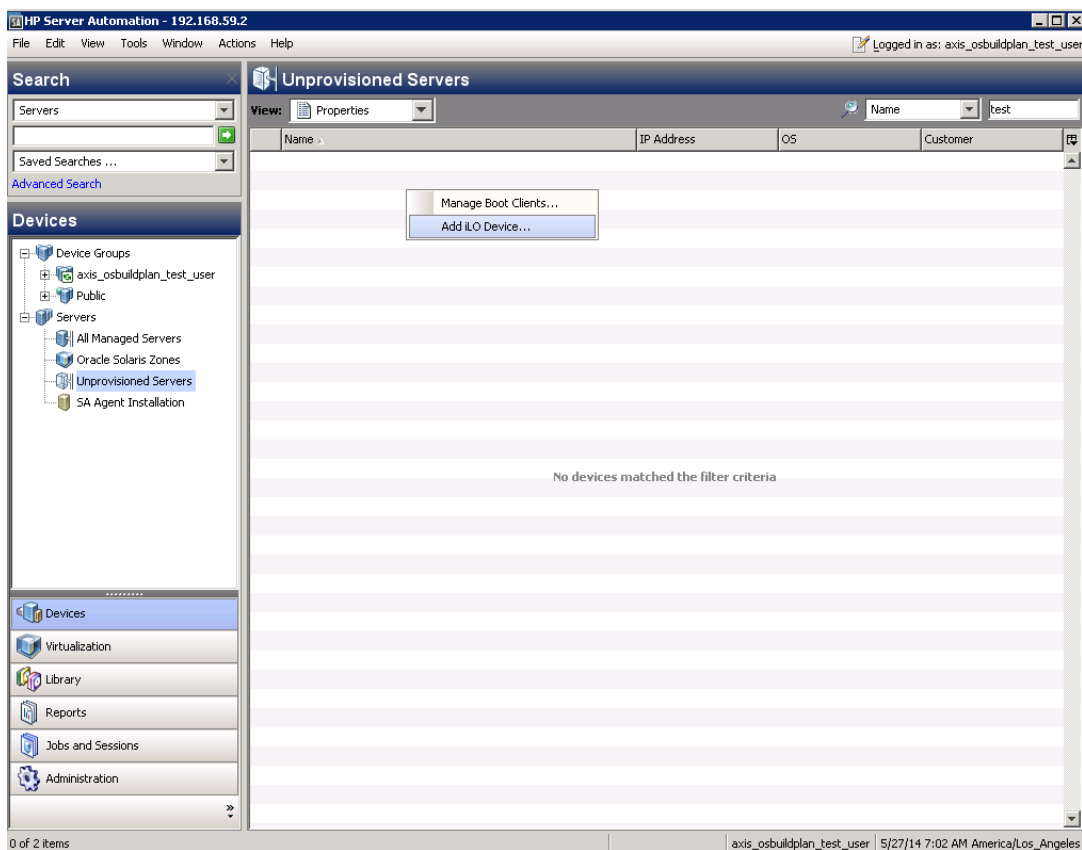
As part of this process, a new iLO user is created, `hp_automatic_integration_user`, with a strong random password. If the iLO is unregistered or the server record is deleted, this iLO user will also be removed (if connectivity to the iLO is still available).

### Manual Registration

To register iLOs manually:

1. Select **Add iLO Device** from the drop-down menu in the SA Client's Unprovisioned Servers list:





2. Specify the iLO IP address or hostname, port, and credentials. You may also need to select a different SA Realm if iLO connectivity is only available to SA Agent gateways from a specific realm. Note also that you can register multiple iLOs if they share credentials.

**Add iLO Devices**

To add a new iLO/Gen8 device to HP SA enter the following information.

iLO IP Addresses/iLO Hostname: *Comma separated list of iLO IP Addresses*

iLO Port (Optional):

User Name:

Password:

Realm:

Current Realm: ZEPPELIN-agents

**Add iLO Devices Results:**

iLO IP Address	Status	Result

Detailed Status:

After iLO registration completes, a new server record is created and associated with an iLO manager. However, a new server record is not be created if an existing server matches the one discovered by the iLO.

For IPv4:

**Add iLO Devices**

To add a new iLO/Gen8 device to HP SA enter the following information.

iLO IP Addresses/iLO Hostname:

iLO Port (Optional):

User Name:

Password:

Realm:

Current Realm: ZEPPELIN-agents

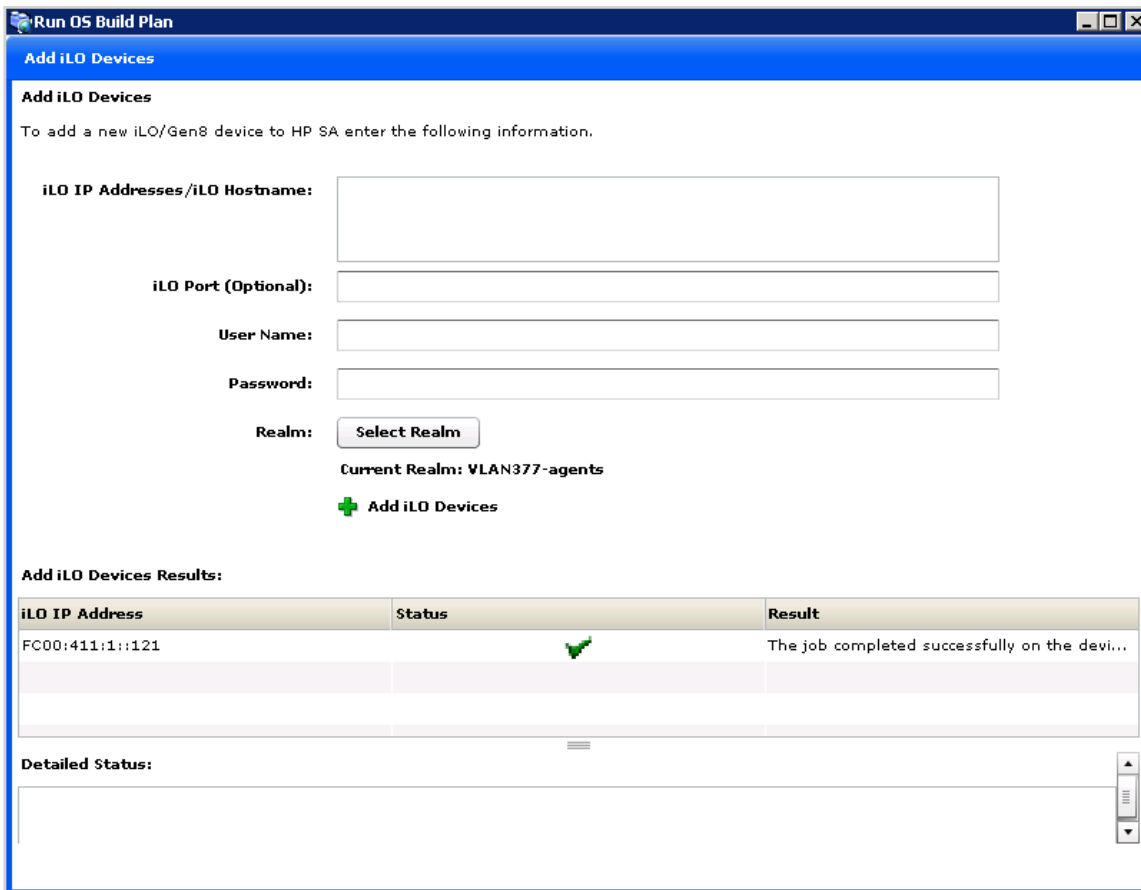
**Add iLO Devices Results:**

iLO IP Address	Status	Result
192.168.244.126	Success	The job completed successfully on the device with id 90001

Detailed status for job:280001

The job completed successfully on the device with id 90001

For IPv6:



See the SA User Guide: Server Automation, “Exploring Servers and Device Groups in the SA Client”.

## Customizing a Target Server for Build Plans

Before you perform SA Provisioning on a target server, you can do some customization for the target server.

### Using Custom Attributes

Custom attributes allow you to control the behavior or outcome of a Build Plan without modifying the SA provided baseline Build Plans.

To see the list of supported Custom Attributes, in the SA Client Navigation pane, set the View to **By Type** and open a Build Plan. In the **Views** pane, select the **Custom Attributes** view to see each set of custom attributes with a blank value. Depending on the customization the attribute targets, you may want to set it to different resources.

Note that a Custom attribute must have a non-blank value in order for it to be considered present on a resource.

During a Build Plan execution, custom attributes are searched in order on the following resources:

- Server
- Device Group
- Customer
- Realm
- Facility
- Build Plan

### Using Device Groups

Device groups allow you to customize a Build Plan run using Custom Attributes on more than one server. SA supports two types of device groups:

- **Dynamic**

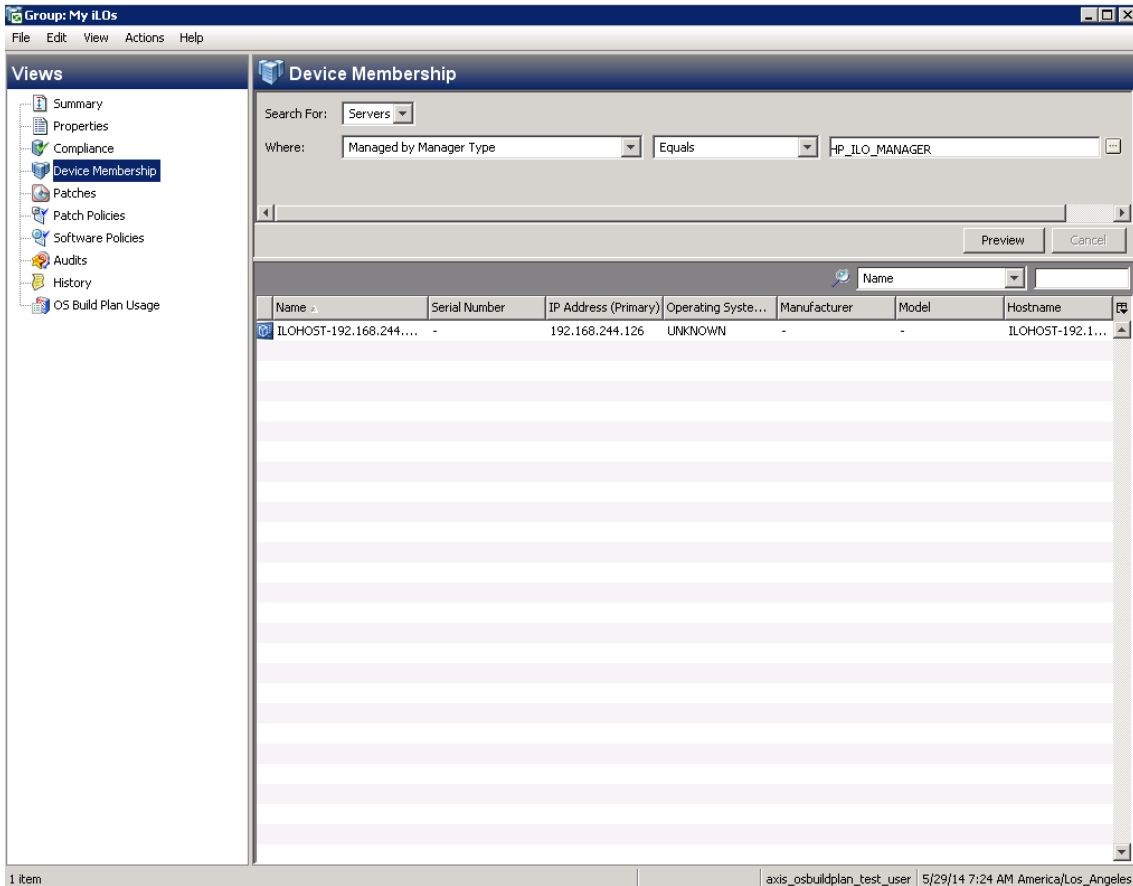
Device membership in a dynamic device group is defined by a device membership policy from the beginning and its group membership is recalculated periodically. See the SA User Guide: Server Automation, “About Dynamic Device Groups”.

- **Static**

Static device group device membership is based on your specification.

Build Plans also support modifying the membership of a static device group using the `Add to Device Group` step. See the SA User Guide: Server Automation, “About Static Device Groups”.

The following example shows a dynamic device group that only targets servers that have an iLO manager:

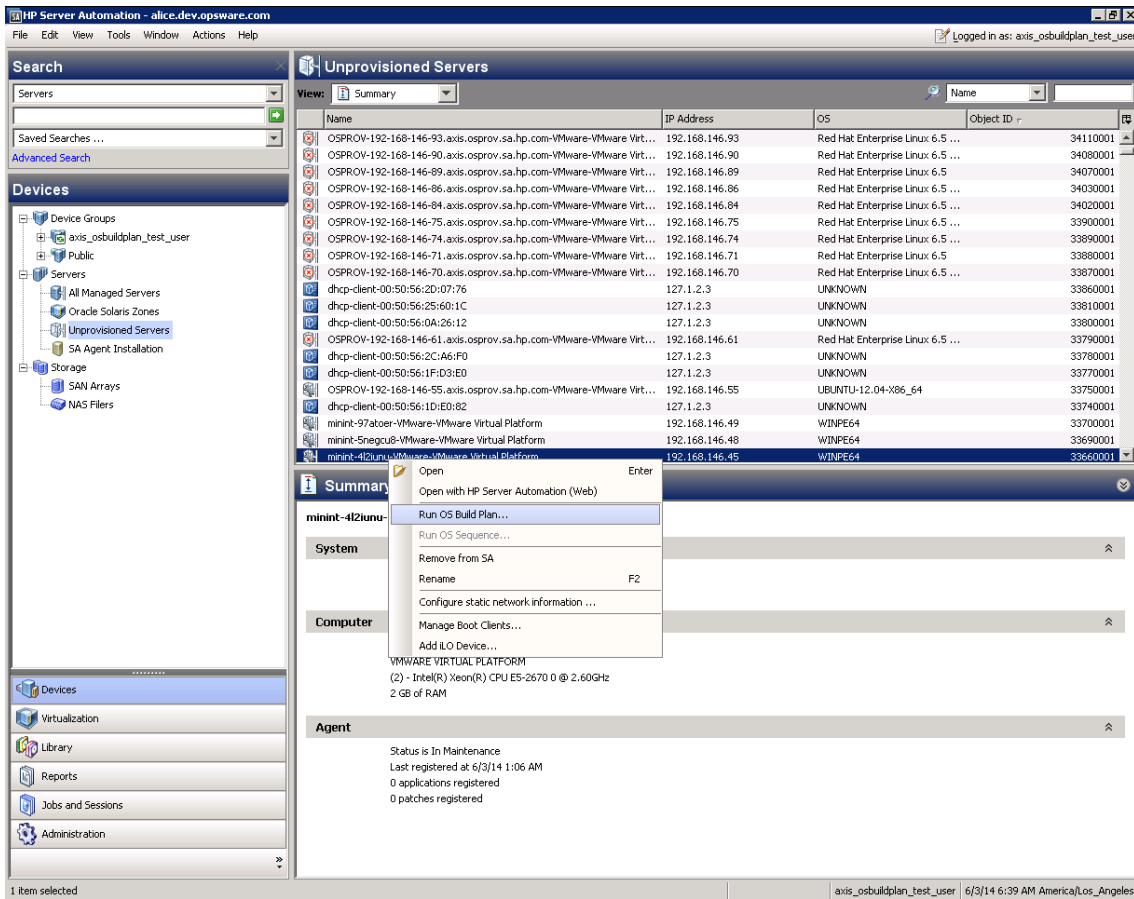


### Phase 3: Running a Build Plan

To provision an operating system on a server or multiple servers using a Build Plan, first log on to the SA Client specifying the SA Core that manages the servers you will install the operating system on. There are several ways to start the provisioning process.

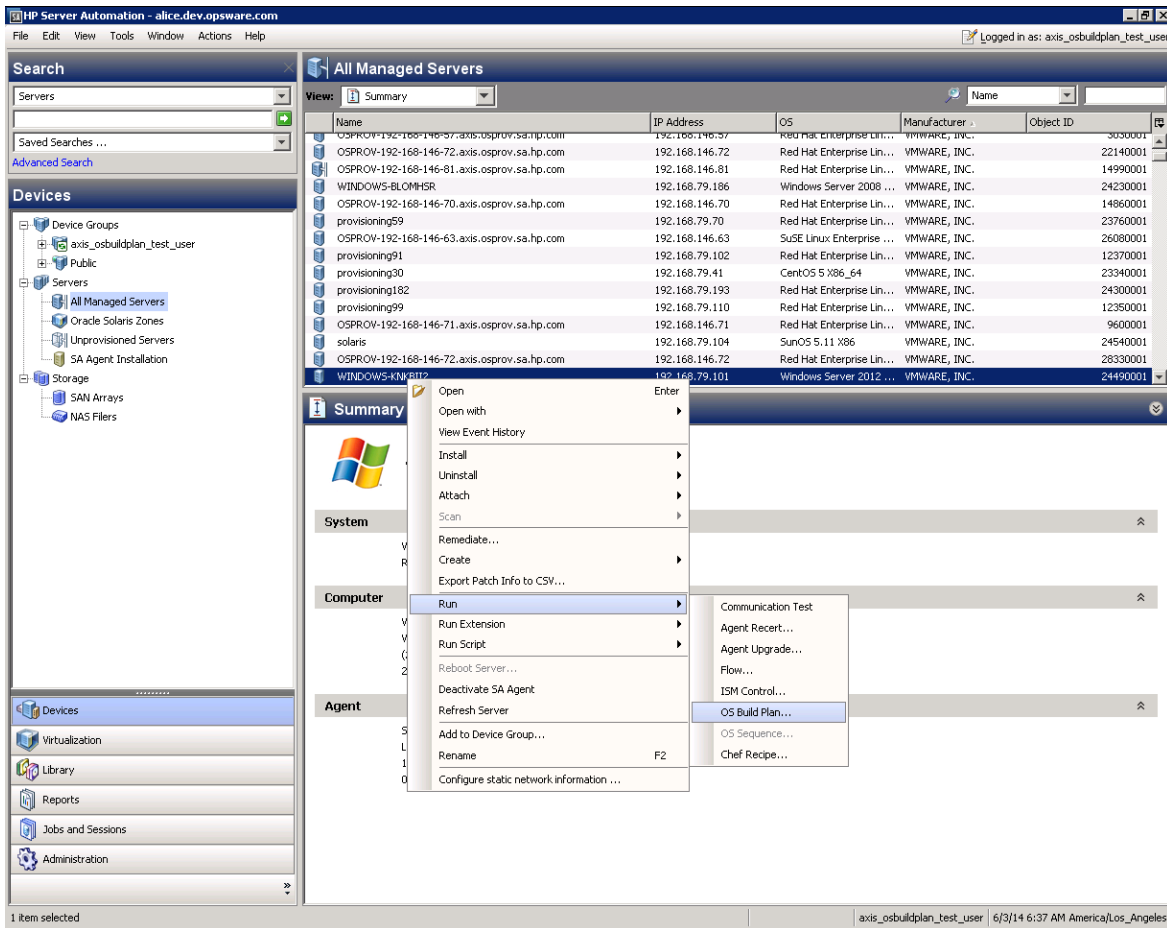
#### Opening the Run Build Plan Wizard from Unprovisioned Servers

1. In the SA Client Navigation pane, select **Devices > Unprovisioned Servers**.
2. Right-click on a listed server and, from the context menu, select **Run > OS Build Plan...**



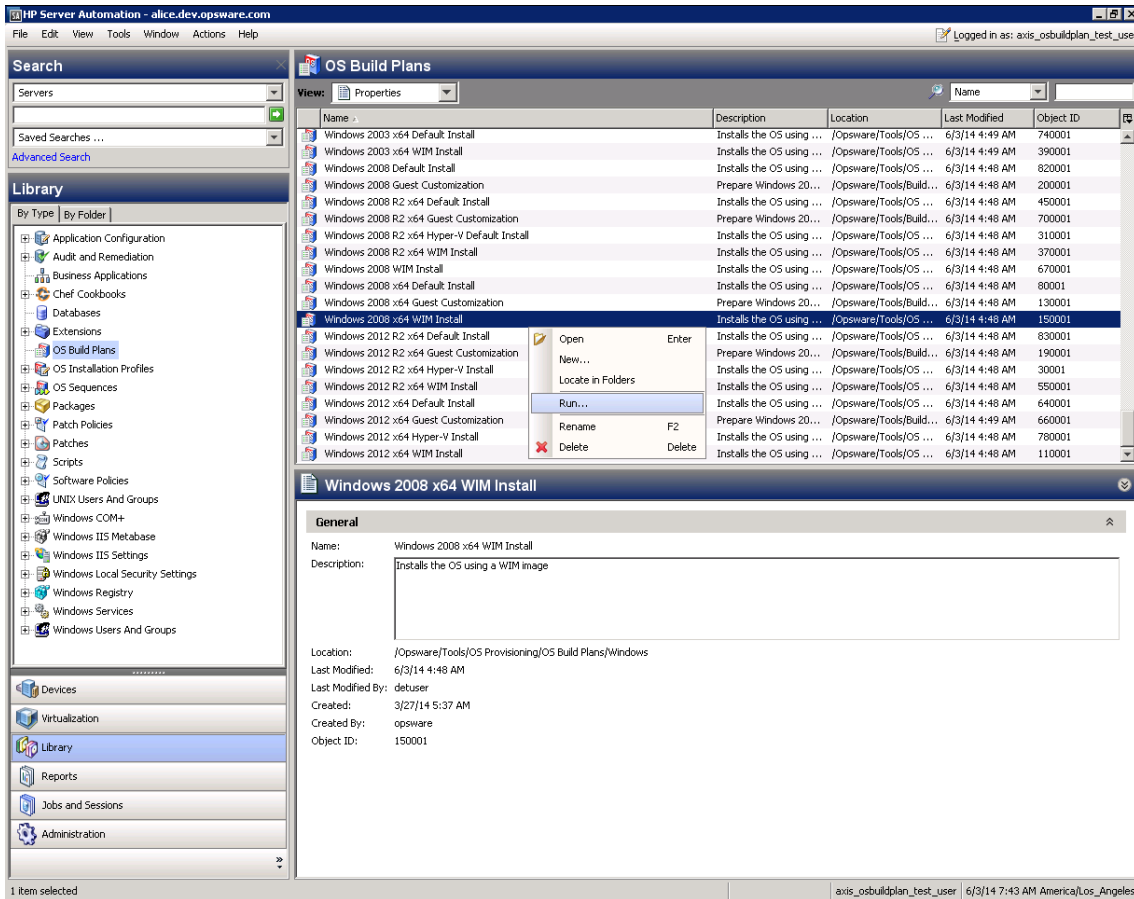
## Opening the Run Build Plan Wizard from Managed Servers

1. In the SA Client Navigation pane, select **Devices > All Managed Servers**.
2. Right-click on a listed server and, from the context menu, select **Run > OS Build Plan...**



## Opening the Run Build Plan Wizard from the SA Client Library

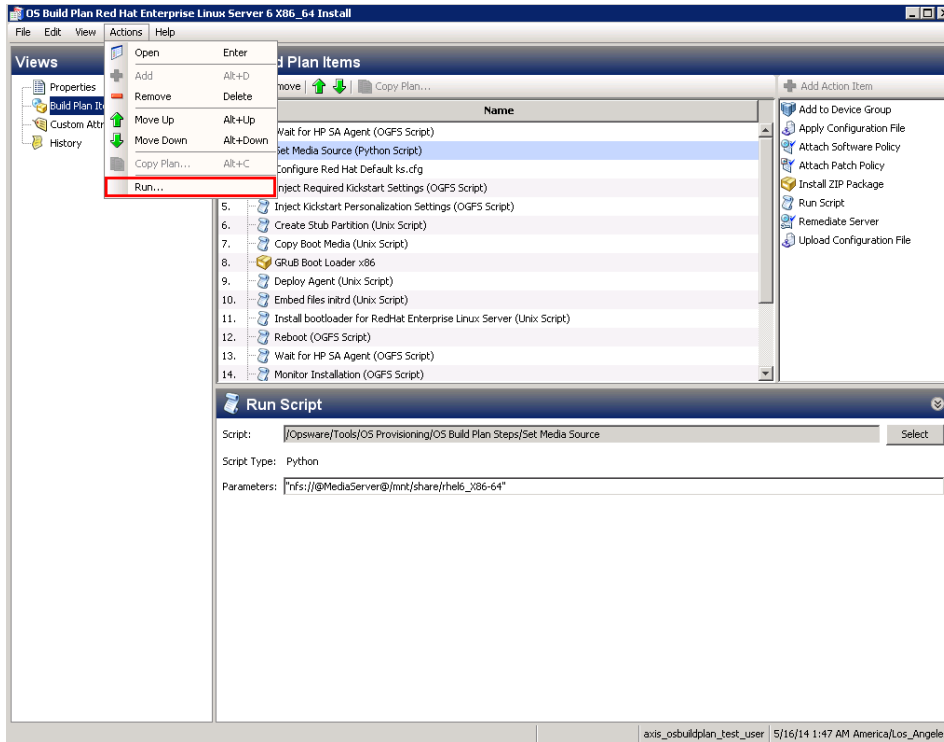
1. In the SA Client Navigation pane, open the **Library** panel and select **OS Build Plans**.
2. Right-click on a listed Build Plan and select **Run**.



### Opening the Run Build Plan Wizard from an Open Build Plan

1. In the SA Client Navigation pane, open the **Library** panel and select **OS Build Plans**.
2. Open the Build Plan.
3. From the **Actions** menu select **Run...**



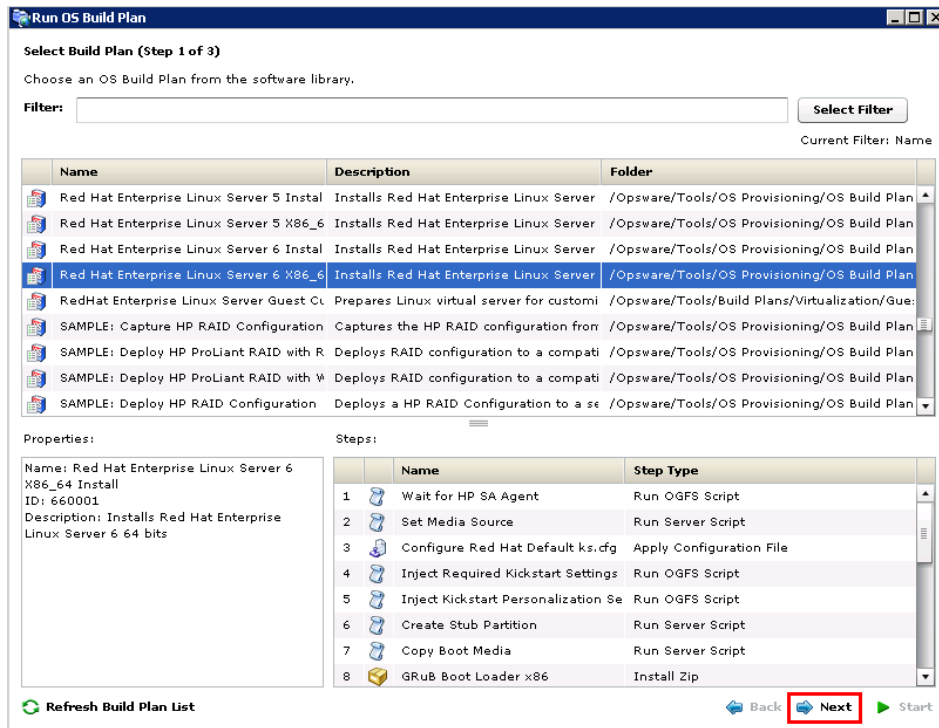


Alternatively, you can select the **Action** menu, then **Run > Build Plan** and choose a target server in the Run Build Plan window or use the search pane to search for a list of Build Plans.

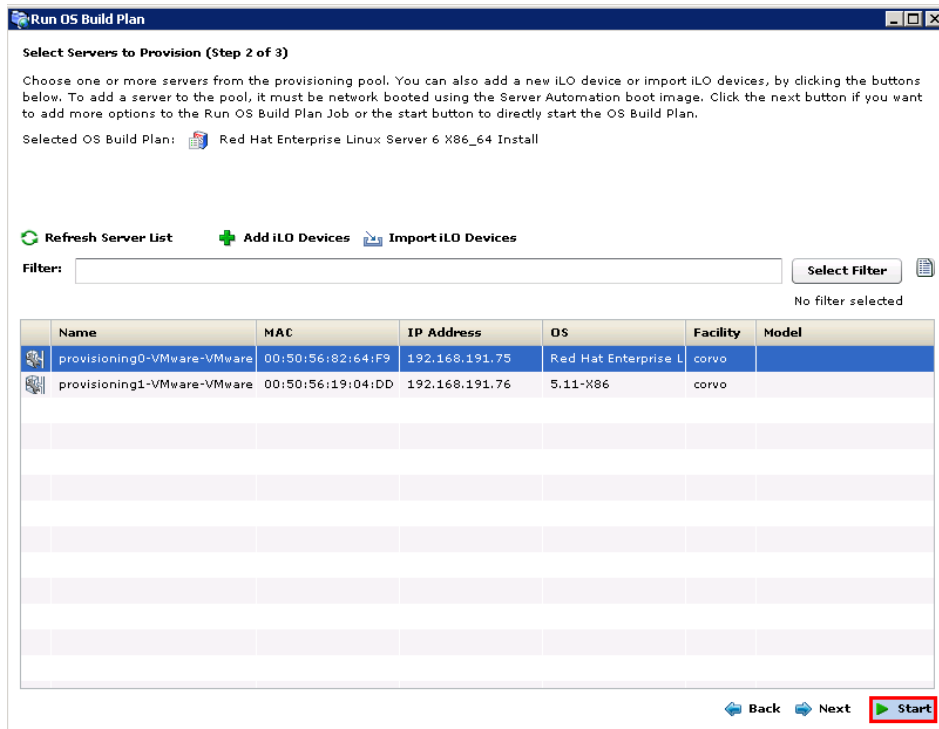
**Note:** If the Run Build Plan option is not available in any of the previous examples, the SA Agent on the targeted devices is not reachable.

## Starting the Build Plan

1. In the Run OS Build Plan window, confirm the selected Build Plan by pressing Next.



2. Select one or more servers on which to run the selected Build Plan.
3. Press Start.



4. The progress of the Build Plan job is displayed.
5. When the Build Plan job completes, the server is up-and-running and under SA management.

### Searching for Active/Completed/Failed Run Build Plan Jobs

1. In the SA Client Navigation pane, select **Jobs and Sessions > Job Logs**.
2. In the Jobs and Sessions list, Build Plan jobs are listed with the job type of Run OS Build Plan with a description that displays the Build Plan name.

Job ID	Type	Description
68850001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_CentOS_6_Static_64x86 for osprovvc001' build plan (id = 48390001) against 1 server(s)
68840001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Oel_5_CD_510x64 for osprovvc001' build plan (id = 48380001) against 1 server(s)
68830001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Oel_5_DHCP_510x64 for osprovvc001' build plan (id = 48370001) against 1 server(s)
68800001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Oel_6_CD_64x64 for osprovvc001' build plan (id = 48350001) against 1 server(s)
68790001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Oel_6_DHCP_64x64 for osprovvc001' build plan (id = 48340001) against 1 server(s)
68720001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Oel_6_Static_64x86 for osprovvc001' build plan (id = 48320001) against 1 server(s)
68710001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_5_DHCP_510x86 for osprovvc001' build plan (id = 48300001) against 1 server(s)
68700001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_5_CD_59x64 for osprovvc001' build plan (id = 48310001) against 1 server(s)
68690001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_5_DHCP_510x86_Http for osprovvc001' build plan (id = 48290001) against 1 server(s)
68670001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_5_OLPers_59x64 for osprovvc001' build plan (id = 48280001) against 1 server(s)
68660001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_5_OldStatic_59x64 for osprovvc001' build plan (id = 48270001) against 1 server(s)
68580001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_6_OLPers_65x64-part-1 for osprovvc001' build plan (id = 48260001) against 1 server(s)
68530001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_6_DHCP_60x64_Workstation for osprovvc001' build plan (id = 48230001) against 1 server(s)
68520001	Run OS Build Plan	Run 'AXIS_O5BP_Linux_Rhel_6_CD_65x86 for osprovvc001' build plan (id = 48240001) against 1 server(s)

3. The Jobs and Sessions screen displays information about each Build Plan job including
  - Job ID
  - Job Type
  - Name of the Build Plan
  - Job Description
  - Number of servers the job was run against

### Personalize Network Settings

SA supports complete network personalization, both during the provisioning process and for the installed operating system. Network settings for existing servers can be also changed.

Network personalization is achieved using the `hpsa_netconfig` custom attribute using simple JSON (<http://json.org/>) syntax to specify the network configuration to be configured on the target system. For example:

```
{
  "hostname" : "testname",
  "domain" : "test.domain.com",
  "workgroup" : "someWorkgroup",
  "interfaces" : [
  {
```

```

"macAddress": "11:22:33:44:55:66",
"enabled": true,
"dhcpv4": true,
"ipv6Autoconfig": true,
"provisioning": true,
"dnsServers" : [ "192.168.0.30", "192.168.0.31", "FC00:2::30",
"FC00:2::31" ],
"dnsSearch" : [ "test.domain.com", "domain.com" ],
"winsServers" : [ "192.168.0.34" ],
"staticNetworks": [
"192.168.0.123/24",
"192.168.0.124/255.255.255.0",
"FC00:2::123/64"
],
"vlanid" : 2,
"ipv4gateway": "192.168.0.1",
"ipv6gateway": "FC00:2::1"
}
],
"virtualInterfaces" : [
{
"interfaceName" : "br0",
}
]
}

```

### Mandatory and Optional Fields

If you do not specify the `hpsa_netconfig` custom attribute, SA automatically determines the interface used by the SA Agent to communicate with the SA Core when the personalization runs. This interface, called the *Provisioning Interface*, is configured automatically through DHCP.

If the `hpsa_netconfig` custom attribute is present and contains *interfaces*, the `macAddress` field defaults to the MAC address of the provisioning interface if one is not present.

Because there is a single provisioning interface, there can only be one interface definition in `hpsa_netconfig` that does not have a MAC address.

MAC addresses are needed to uniquely identify the server's network interfaces. All other fields are optional and have default values.

The `hpsa_netconfig` format does not make any assumptions about how the networks to which the server is joined are configured. For this reason, only minimal validation is performed. SA does not verify that the settings lead to valid connectivity between the SA Agent and the SA Core. You should verify that the specified network settings will allow the SA Agent to connect to the SA Core after they are applied. Other obvious error cases, such as disabling the provisioning interface, are validated.

## Description of Individual Fields

### **enabled**

The **enabled** value handles the state in which the interface will be after the network is configured. If the value is *false*, the interface will still be configured as intended, but it will be deactivated.

### **hostname, domain**

The host name (also known as the *computer name*) is used to identify the node on the network. The domain name is the DNS registered domain of the server. Together they account for the fully qualified domain name (FQDN) of the server.

### **interfaces**

A list of the system's physical network interfaces that are to be configured. Each interface (as identified by the MAC address) can have a single entry in the list.

### **macAddress**

The Media Access Control (MAC) address of the network interface. Multiple formats are accepted, colon or dash separated, or just a string of hexadecimal numbers.

### **dhcpv4**

Controls the use of DHCP for acquiring IPv4 network addresses.

### **ipv6Autoconfig**

Controls the use of IPv6 stateless address autoconfiguration (SLAAC) and DHCPv6 simultaneously. The IPv6 router should be configured to advertise DHCPv6 configuration. If not specified, depending on how the SA Agent is connected to the SA Core, it will be set to true (an IPv6 connection) or false (an IPv4 connection).

### **provisioning**

This field is used to explicitly specify the interface to be used for provisioning. Only one provisioning interface is supported. Use of this field is not recommended outside of complex scenarios. In most cases, SA will be able to (and will) configure this automatically.

### **dnsServers, dnsSearch, winsServer**

Controls the name resolution settings. The order in which the values are specified will be the order of configuration. The first `dns nameserver`, `dns domain` or `winserver` in the list will be the primary selection. DNS servers can be a combination of both IPv4 and IPv6 addresses. For WINS servers only, IPv4 addresses are supported.

### **staticNetworks**

A list of static networks to configure on the interface. IPv4 addresses can use the CIDR notation, or IP address / network mask notation. IPv6 addresses will use the IP address / prefix length notation. The first address in the list will be the first one to be applied.

### **ipv4gateway/ipv6gateway**

The IP version 4 default gateway or IP version 6 (next hop) address. <<**Could these be changed to IPv4 and IPv6 here?>>**

### **vlanid**

The VLAN ID used to tag packets for this interface.

### **virtualInterfaces**

This section configures the non physical interfaces. These are not identified by their MAC address but by their `interfaceName`. The virtual interfaces are configured similarly to the physical ones (using fields as `dhcpv4`, `staticNetworks`, etc.)

### **interfaceName**

Identifier for the configured virtual interface. This field is not necessary for the physical interfaces which are identified by their MAC address.

## **Where is hpsa\_netconfig Used?**

You can personalize the network settings at different stages of the provisioning process. The network can be personalized across all these stages, or selectively. For example it can start out with DHCP-based provisioning and switch to static networking after the system is provisioned.

### **Service OS with Personalized Network Settings**

Servers can be brought into Maintenance mode using the static provisioning images, that provide a boot configuration screen used to configure these network settings. In this case, the boot configuration screen configures `hpsa_netconfig` to be used throughout the provisioning process. The network settings you specify in the boot configuration screen are applied to the final installed OS as well.

For HP ProLiant Gen8 servers, you can also use the embedded service OS for PXE-less and DHCP-less provisioning, by registering the server with its iLO address and setting the `hpsa_netconfig` custom attribute. In this case, the MAC address is mandatory, since the SA Agent has not reported hardware information to the SA Core yet so the MAC address can not be determined automatically.

The ability to configure the service OS statically allows you to provision without DHCP and network boot infrastructure (PXE and TFTP server).

### During the Provisioning Process

Using the information from the `hpsa_netconfig` custom attribute, SA automatically injects the required settings in the OS *installation profile* for the provisioning interface to run the vendor OS installer with the specified network settings. This is the only interface configured this way, and only one IP address will be configured for it. Depending on whether the Service OS used is IPv4- or IPv6-based, the OS installer will have an IPv4 or IPv6 address as configured in `hpsa_netconfig` for the provisioning interface.

For the OS installer on IPv6, although the protocol supports multiple IPs on the same interface, you cannot have static and dynamic (SLAAC and DHCP) IPs at the same time. If you have both static and dynamic (`ipv6Autoconfig`) defined in `hpsa_netconfig` for the provisioning interface, only the static IP settings will be used.

This configuration only allows SA to install in an environment that does not use DHCP. The network configuration is completed after the OS is installed.

### Network Personalization of an Installed System

The complete configuration of all network interfaces and all addresses (IPv4 and IPv6) can be accomplished using **Personalize Network Settings of Installed System** script as a **Run Script** step.

This step already exists in most baseline Build Plans. It can be added to any Build Plan or you can create a separate Build Plan intended only for network configuration.

Because this step can lead to loss of connection to the SA Agent, it must always be followed by a **Wait for HP SA Agent** step.

Running the personalization step configures the targeted device in a series of steps: updating the computer name (`hostname`), the domain and DNS information and the specified network settings. After updating the network settings for a persistent configuration, a network stack reset is executed. Then the SA Agent is forced to report the new hardware changes. All changes are platform-specific, as the **Personalize Network Settings of Installed System** step can detect the targeted platform.

Applying the new configuration must handle different possible scenarios. Updating the DNS and domain information means reorganizing the previous configuration so that the new configuration will act as the primary (for example, setting the new DNS IP as the first `nameserver` in `/etc/resolv.conf`). Updating the network settings requires clearing up the old configuration (only the fields that are handled by the `hpsa_netconfig` custom attribute) before committing to the new configuration, clearing and creating aliases if needed (one of the cases would be if multiple static IPv4 networks are specified), and enabling or disabling the dual-stack network interface.

**Note:** In all scenarios, the old configuration is preserved if you do not intentionally modify it.

### Example:

Assume that the device already has three configured interfaces ("`eth0`", "`eth1`" with "`eth1:1`" alias, "`eth2`"). If the `hpsa_netconfig` custom attribute is set to configure "`eth0`" to static with an alias and "`eth1`" to dhcp, the step applies the personalization to "`eth0`", creating the

alias, and to "eth1", deleting the existing alias (the result being: "eth0" with "eth0 : 1" alias, "eth1", "eth2"). Other interfaces will remain unchanged (in this case, "eth2").

#### **Red Hat Enterprise Linux, CentOS, Oracle Enterprise Linux Platform**

- Updated configuration files for the new computer name : `/etc/hosts`, `/etc/sysconfig/network`.
- The `hostname` command is also executed for the runtime configuration.
- Updated configuration files for the domain and DNS information: `/etc/resolv.conf`.
- Updated configuration files for the network specific configuration : `/etc/sysconfig/network` and `/etc/sysconfig/network-scripts/ifcfg-ethXX`.
- Restarting network : `/etc/init.d/network restart` (this step could lead to loss of connection to the SA Agent).

#### **Ubuntu Platform**

Network configuration does not support mapping in `/etc/network/interfaces` because `hpsa_netconfig` does not support multiple naming. Example of unsupported configuration:

```
mapping eth0
script /usr/local/sbin/map-scheme
map HOME eth0-home
map WORK eth0-work
iface eth0-home inet static
address 192.168.1.1
netmask 255.255.255.0
up flush-mail
iface eth0-work inet dhcp
```

- Updated configuration files for the new computer name : `/etc/hosts`, `/etc/hostname`.
- Updated configuration files for the domain and DNS information: `/etc/resolv.conf` and `/etc/resolvconf/resolv.conf.d/original`
- Updated configuration files for the network specific configuration : `/etc/network/interfaces`
- Restarting network : `/etc/init.d/networking restart` (this step could lead to loss of connection to the SA Agent)

#### **SUSE Platform**



- Updated configuration files for the new computer name : `/etc/hosts` and `/etc/HOSTNAME`.
- The `hostname` command is also executed for the runtime configuration.
- Updated configuration files for the domain and DNS information: `/etc/resolv.conf` and `/etc/sysconfig/network/config`.
- Updated configuration files for the network specific configuration : `/etc/sysconfig/network/routes`, `/etc/sysconfig/network/ifcfg-ethXX` and `/etc/sysctl.conf`.
- Restarting network : `/etc/init.d/networking restart` (this step could lead to loss of connection to the SA Agent).

### VMware ESXi Platform

**Note:** Network configuration is not supported for this platform because, after installation, the ESXi OS is agentless. Static network configuration is applied during installation by injecting it into the Kickstart file.

### Windows Platforms

The same configuration process can be used to apply persistent settings or to configure the device when running the installer.

Applying personalization implies executing a series of Windows-specific commands that configure the system with the specified information.

This also means that the configuration is visible immediately without the necessity to apply it after reboot.

**Note:** The device must be rebooted so that the changes to the hostname (computer name) are visible to the OS.

### Personalize Network Settings of Installed Systems Build Plan

The Build Plan is created especially for network personalization of installed systems. It requires the SA Agent to be in production after installation.

The network personalization is done as described in the sections above.

The Build Plan contains the **Skip steps based on Custom Attribute** step. This step allows the Build Plan to take advantage of the *Flow Control* feature to skip or execute the **Reboot** step. It has a predefined custom attribute: `skip_reboot`, with a default value of `no` and, as argument, the number of steps to skip. If the value of `skip_reboot` is set to `yes`, the **Reboot** step is not executed. See also [Flow Control Mechanism](#).

#### Example 1

A device has three configured interfaces (`eth0`, `eth1` with `eth1:1` alias, `eth2`).

If `hpsa_netconfig` is set to configure `eth0` to static with an alias and `eth1` configured to DHCP, the personalization step applies the personalization to `eth0` creating the alias and to `eth1` deleting the existing alias (the result being: `eth0` with `eth0:1` alias, `eth1`, `eth2`). Other interfaces remains unchanged (in this case, `eth2`).

*Red Hat Enterprise Linux, CentOS, Oracle Enterprise Linux Platforms*

- **Updated configuration files for the new computer name:** `/etc/hosts`, `/etc/sysconfig/network`
  - the `hostname` command is also executed for the runtime configuration
- **Updated configuration files for the domain and DNS information:** `/etc/resolv.conf`
- **Updated configuration files for the network specific configuration:** `/etc/sysconfig/network`, `/etc/sysconfig/network-scripts/ifcfg-ethXX`
- **Restarting network:** `/etc/init.d/network restart`
  - this step could lead to loss of connection to the SA Agent

#### *Ubuntu Platform*

Network configuration does not support mapping in `/etc/network/interfaces` because `hpsa_netconfig` does not support multiple naming. Example of unsupported configuration:

```
mapping eth0
```

```
script /usr/local/sbin/map-scheme
```

```
map HOME eth0-home
```

```
map WORK eth0-work
```

```
iface eth0-home inet static
```

```
address 192.168.1.1
```

```
netmask 255.255.255.0
```

```
up flush-mail
```

```
iface eth0-work inet dhcp
```

- **Updated configuration files for the new computer name:** `/etc/hosts`, `/etc/hostname`
  - the `hostname` command is also executed for runtime configuration
- **Updated configuration files for the domain and DNS information:** `/etc/resolv.conf`, `/etc/resolvconf/resolv.conf.d/original`
- **Updated configuration files for the network specific configuration:** `/etc/network/interfaces`
- **Restarting network:** `/etc/init.d/networking restart`
  - Step could lead to loss of connection to the SA Agent

#### *SUSE Platform*

- Updated configuration files for the new computer name: `/etc/hosts`, `/etc/HOSTNAME`
  - The `hostname` command is also executed for the runtime configuration
- Updated configuration files for the domain and DNS information: `/etc/resolv.conf`, `/etc/sysconfig/network/config`
- Updated configuration files for the network specific configuration : `/etc/sysconfig/routes`, `/etc/sysconfig/network-scripts/ifcfg-ethXX`, `/etc/sysctl.conf`
- Restarting network: `/etc/init.d/networking restart`
  - Step could lead to loss of connection to the SA Agent

#### *VMware ESXi Platform*

Network configuration is not supported for this platform because, after installation, the ESXi OS is agentless. Static network configuration is applied during installation by injecting it into the kickstart file.

#### *Windows Platforms*

The same configuration process can be used to apply persistent settings or to configure the device when running the installer. Applying personalization implies executing a series of Windows specific commands that configure the system with the specified information. This also means that the configuration is visible immediately without the necessity to apply it after reboot.

The device must be rebooted so that the changes to the `hostname` are visible to the OS.

## **Extending Windows Hardware Support**

SA Windows Service OS images are bundled with drivers for most common devices. However, there will be cases when new hardware requires new drivers. The procedure below describes how to add new drivers to existing Windows Service OS images.

Decide which Windows Service OS you will upgrade. For example:

- WinPE2.1 32-bit (built from Windows Server 2008/Windows Vista)
- WinPE3.1 64-bit (built from Windows 7 SP1/Windows Server 2008 R2)
- WinPE4.0 64-bit (built from Windows 8/Windows Server 2012)
- Make a backup of the image and copy it to a Windows machine where you have installed the latest WADK (<http://technet.microsoft.com/en-us/library/hh824947.aspx>).
- Obtain the required drivers. For example, if you want to update a WinPE3.1 based image, look for drivers for Windows 7 or Windows Server 2008 R2.
- Use the procedure described on the Microsoft page *Add Device Drivers on an Offline Windows PE Image* ([http://technet.microsoft.com/en-us/library/dd799289\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/dd799289(v=ws.10).aspx)).
- Replace original image with the modified image.

## Joining a Windows Domain or Workgroup

Windows Build Plans also provide the ability to join a provisioned machine into a *Windows Domain* or *Workgroup*. To do this, use the step called: **Inject Windows Domain or Workgroup Personalization Settings**.

To use this step and join the machine to a Windows Domain, you must use the following custom attributes:

- **DomainName**: the name of the domain that you want the server to join.
- **DomainUser**: the user name which has the permissions to add the server to a domain.
- **DomainPassword**: the password of the `DomainUser`.

To join the server to a Workgroup, you must use only one custom attribute:

- **WorkGroup**: the workgroup name.

To use this step in new Build Plans, do the following:

- Ensure that this step is added after the **Configure Windows Default Unattend.xml**, **Inject Required Unattend.xml Settings** and **Inject Personalization Settings** steps.
- Ensure that this step is added before the **Run Windows Setup** step.
- Use Custom Attributes, if necessary, as described in [Defining Custom Attributes](#).

## Provisioning an Already Provisioned Server

Build plans support re-purposing (reprovisioning) a server. You can do this either fully automatically or manually.

**Note:** Reprovisioning can cause data loss on the server and decommissions the server in SA. For more information on the effects of decommissioning, see the SA User Guide: Server Automation, “Deactivating a Server”.

### Automatic Reprovisioning

To reprovision a server automatically, you use a Build Plan such as `SAMPLE: Prepare server for reprovisioning to Linux` to prepare the server. You run this Build Plan on the *managed server*. It decommissions the server and brings it into Maintenance mode. After the Build Plan job completes, you can use any baseline SA Build Plan to provision to a new OS.

For example, to reprovision the server with Windows, use a Build Plan based on `SAMPLE: Prepare server for reprovisioning to Linux`, editing the parameters of the boot step to prepare the server before starting the Windows provisioning Build Plan.

If you expect to be executing reprovisioning jobs often, you may want to create a Build Plan that specifies both the reprovisioning steps and the steps that provision to a new OS.

## Manual Reprovisioning

SA supports manual preparation of servers for reprovisioning (for example, to boot using a boot image in a DHCP-less environment). Before you can do this, you must manually decommission (deactivate) the server from SA before booting it into Maintenance mode, see the *SA Installation Guide*, “Uninstalling an SA Core”.

Failure to decommission the server will lead to the server failing to boot into Maintenance mode (due to a registration error with the SA Core).

**Note:** Under certain platforms, a server can boot into Maintenance mode without being decommissioned, however, this capability was designed for maintenance tasks only, not reprovisioning, therefore a failure to decommission the server will lead to an inconsistent representation of the server in SA. For example, old software policies might still be attached.

The Decommission Server procedure cannot be used in this setup because it disables the SA Agent, so the step is not able to continue and will fail in the subsequent `wait for HP SA Agent` step or any step that requires the SA Agent to be active.

## Device Naming

Device Naming allows you to customize the name of a server when the server is not yet registered with the SA Core or when you first boot a server into a Maintenance mode OS.

You can also modify the server's name from the SA Client which will take precedence over the Device Naming function. You can do this by specifying a set of rules to match certain device properties. This allows you to have different names based on the device information registered by the SA Agent.

The rules must be added to the `device_discovery_naming_rules` custom attribute associated with the Server's assigned Facility.

The rules are in the form of pairs of regular expressions and templates like:

```
REGEX1 :=TEMPLATE1
```

```
REGEX1 :=TEMPLATE2
```

Where each REGEX tries to match a string constructed from the SA Agent reported information.

When a match is found, the TEMPLATE is applied to construct the server's name.

The TEMPLATE follows Python 2.7 format string conventions, where given a dictionary in the form of

```
{key: [ {'inner key': 'value1', value2}, ...]}
```

you are able to reference the values as follows:

```
value1 = {key[0][inner key]}
```

```
value2 = {key[1]}
```

The dictionary used for the TEMPLATE is constructed by the SA Agent during hardware registration. The following fields are relevant:

- `chassis_id` - chassis ID
- `dvc_id` - server SA object ID, machine ID
- `dvc_mfg` - server's manufacturer
- `dvc_model` - server's model interfaces\[0\]\[hw\_addr\] - the MAC address of the first reported network interface
- `os_version` - SA Agent reported OS version
- `serial_num` - serial number
- `server_location` - server's location (only available for HP ProLiant blades)
- `system_name` - hostname
- `uuid` - UUID

The following is a simple example:

```
. *server_location.* ::= {system_name} - {server_location[rack]} -
{server_location[enclosure]} - {server_location[bay]}
```

This specification matches any server that reported a location and sets the name to the hostname followed by rack, enclosure and bay.

The following is a slightly more advanced example:

```
. *ProLiant BL.* ::= {server_location[rack]} - {server_location
[enclosure]} - {server_location[bay]}
. *ProLiant* ::= {serial_num}
. *VMware.* ::= {system_name} - {interfaces[0][hw_addr]}
```

The first line of this example matches only *ProLiant BL* servers and assigns a name. The second line is needed only when the first line finds no matches and finds HP ProLiant other than the BL model. The third line explicitly matches only VMware hardware and assigns the hostname followed by the hardware address of the first network interface.



# SA Provisioning Common Use Cases

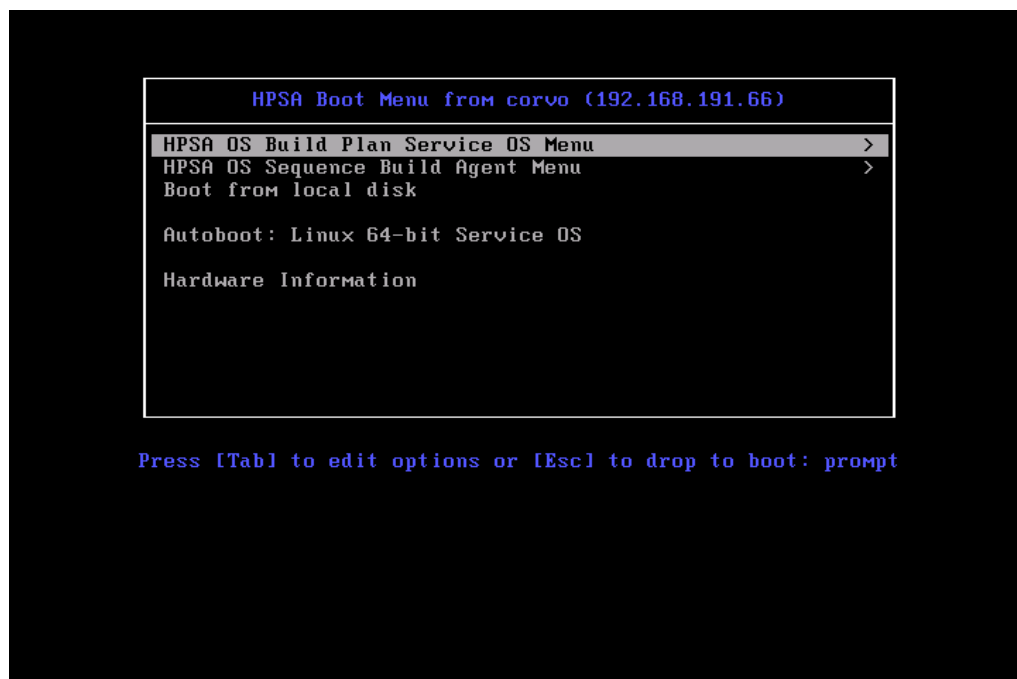
**Note:** Before attempting the following use cases, you must have configured DHCP on the SA Core. See the [DHCP Configuration \(IPv4 and IPv6\) for SA Provisioning](#).

After you are familiar with [SA Provisioning Basics](#), you are ready to start provisioning. The following are some examples that will help you get started.

## Provisioning Windows-Based Servers

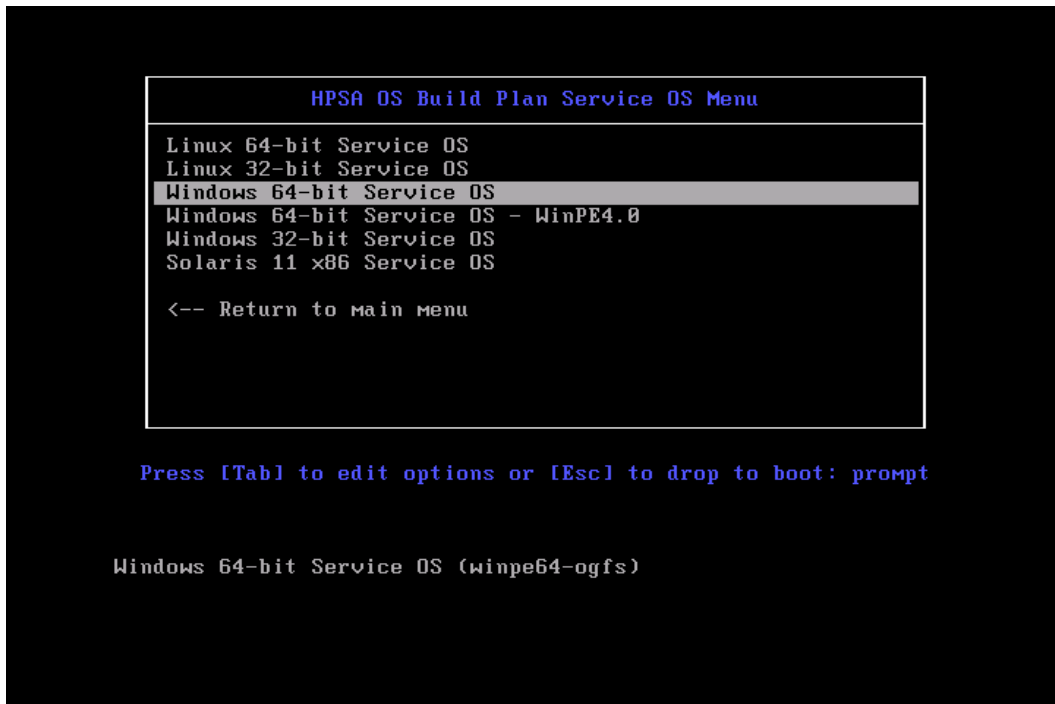
*Provisioning Windows Server 2008 R2*

1. Network boot a server as described in [Network Booting](#).
2. On the network boot menu, select HPSA OS Build Plan Service OS Menu.

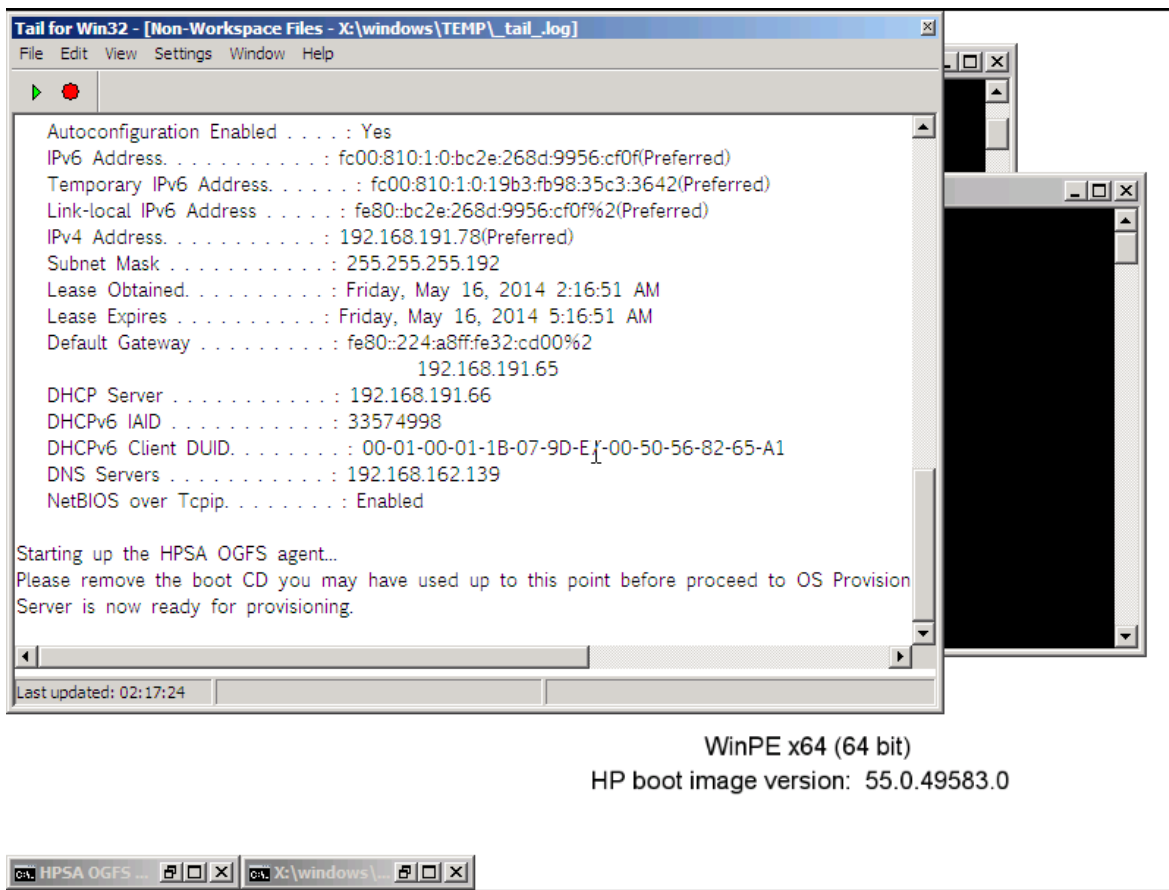


3. Select Windows 64-bit Service OS.

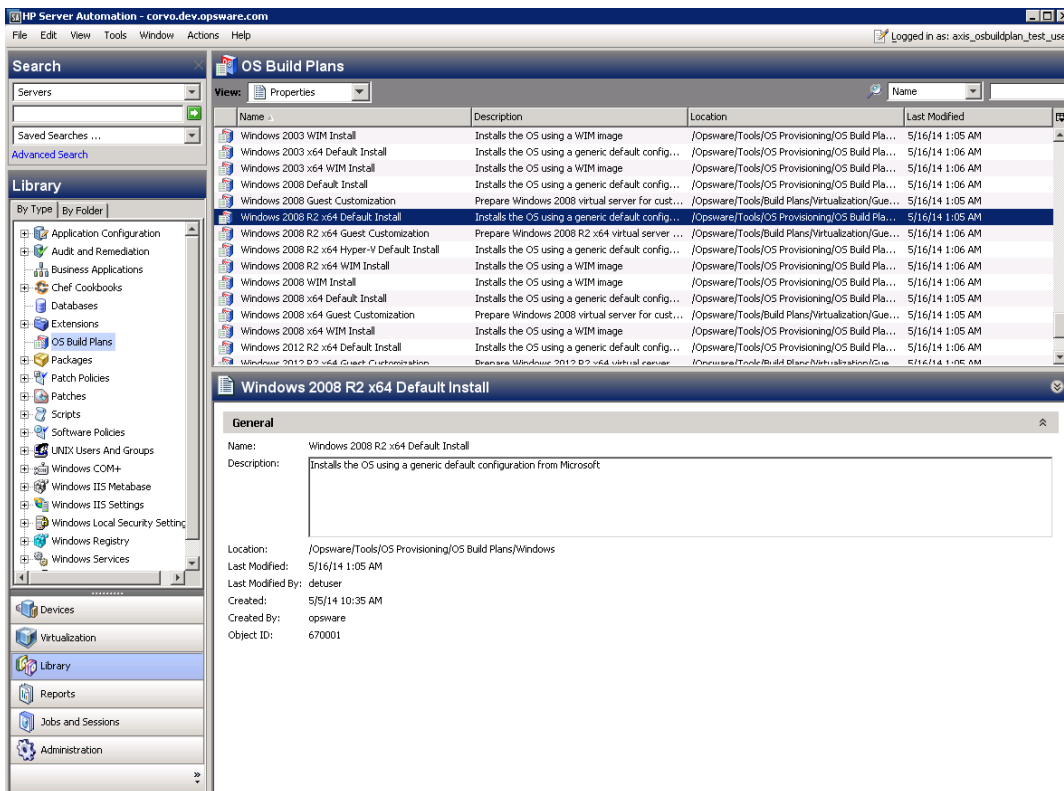




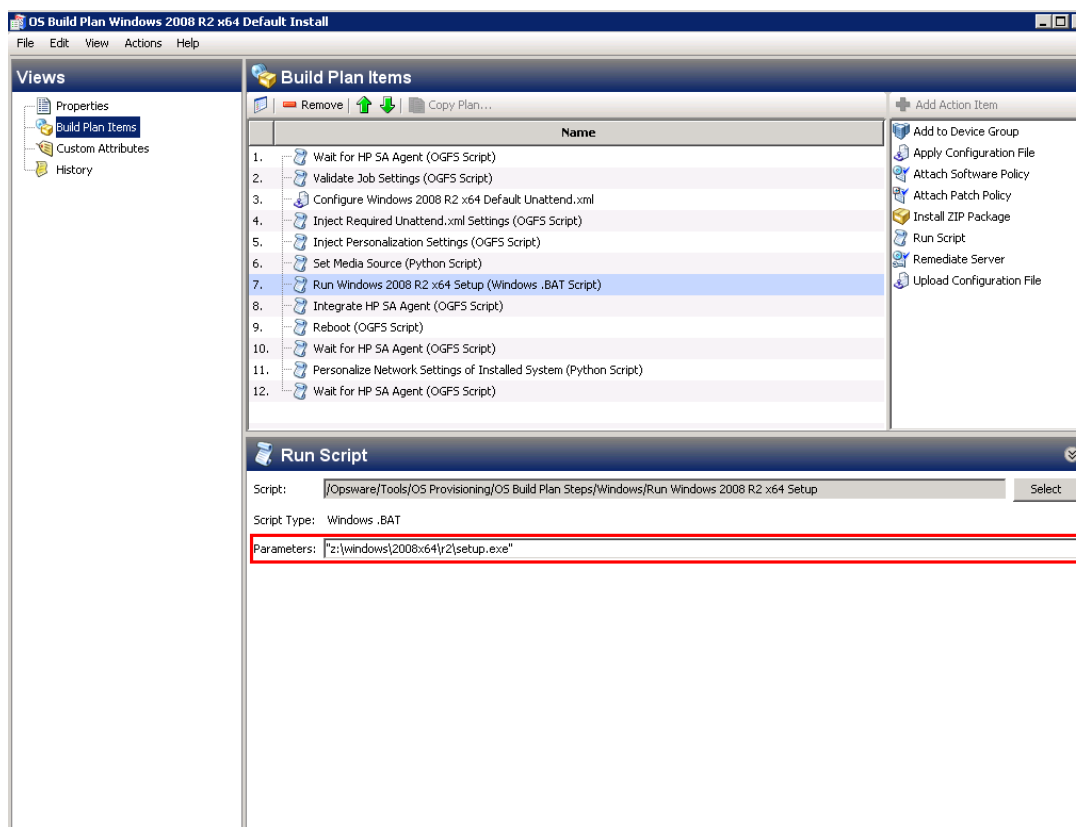
4. Wait for the server to enter Maintenance mode.



5. Note the server's IP address; this will help you identify the machine later on.
6. Start the SA Client and in the Navigation pane, select Library, set the view to By Type and select **OS Build Plans...**
7. From the displayed list of available Build Plans, select and open the Build Plan, Windows 2008 R2 x64 Default Install.



8. In the Build Plan Edit window, from the **Views** panel tree, select **Build Plan Items**. A list of Build Plan steps is displayed.
9. From the listed steps, select **Set Media Source**. The Run Script panel displays the Parameter setting for this step. Note the default protocol, **SMB**, and the media path `/osmedia`.
10. Select the **Run Windows 2008 R2 x64 Setup** Build Plan step
11. In the **Parameters** field, note the path under which ``setup.exe`` is expected to be found, `z:\windows\2008x64\r2\setup.exe`.

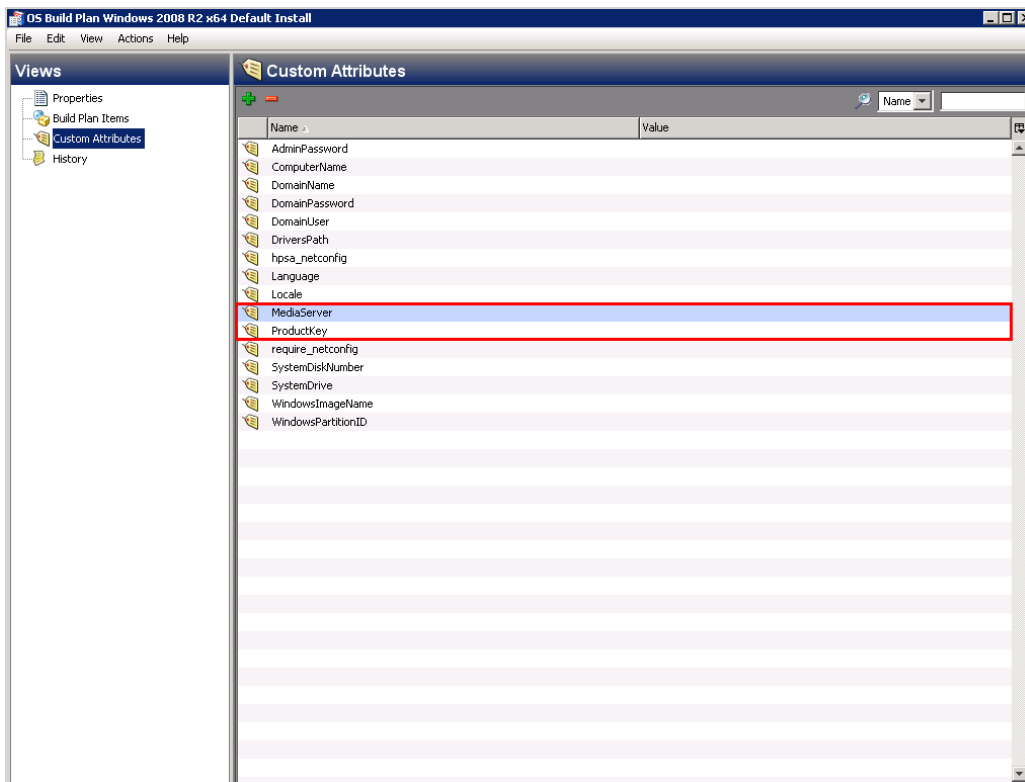


12. Prepare a media server and a SMB share with the Windows Server 2008 R2 ISO unpacked under:

`/osmedia/windows/2008x64/r2/`

a concatenation of the **Set Media Source** and **Run Windows 2008 R2 x64 Setup** parameters.

13. In the Edit Build Plan **Views** pane, select **Custom Attributes**.



14. Complete the **MediaServer** and **ProductKey** custom attributes as follows:
- **MediaServer** is the hostname or IP address of the media server.
  - **ProductKey** must be a valid Microsoft Windows Server 2008 R2 product key.

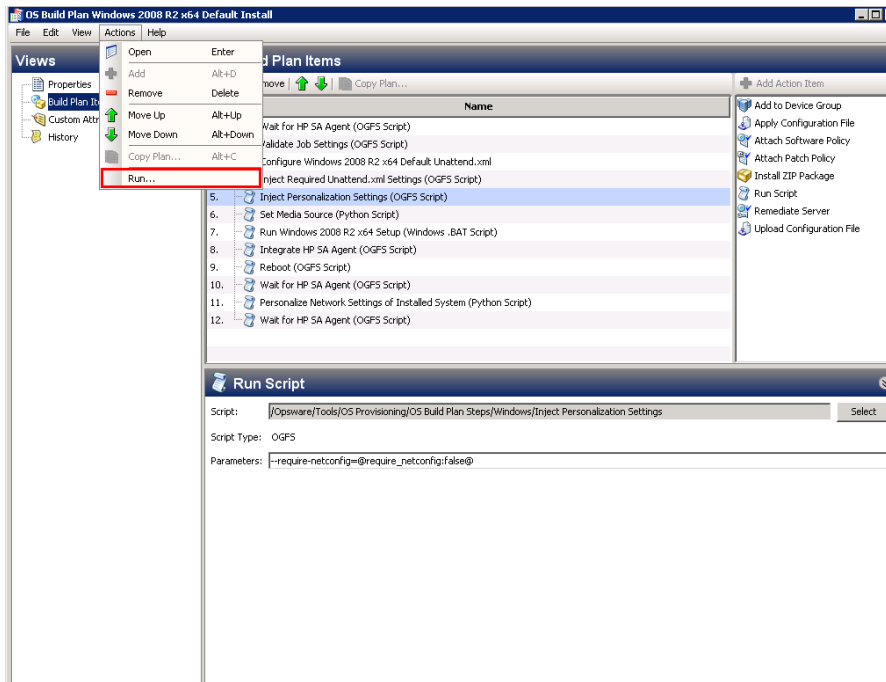
Optional: If you want to add an additional driver search path, specify the **Driver-Path** custom attribute field as well.

Optional: When installing in a language/locale other than US English, in addition to selecting an appropriate media for installation, two custom attributes are defined in the default `unattend.xml` for convenience:

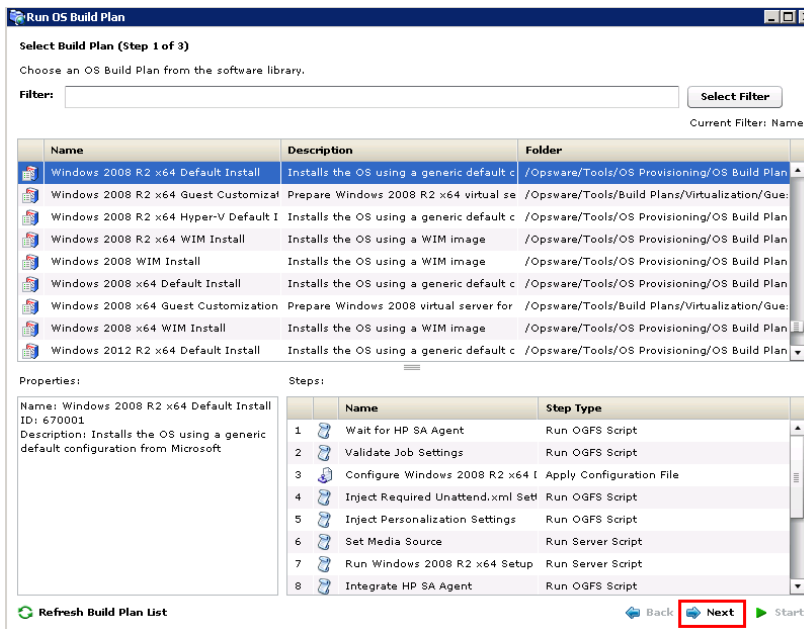
Language and Locale.

- Language: specifies the display language of the installed system
- Locale: specifies the locale (time and currency format, keyboard maps, etc.) of the installed system
- Both custom attributes are in IETF language tag format (for example, `en-US` for US English, `ja-JA` for Japanese, etc.)

15. Save the Build Plan by pressing **Ctrl+S** or **File > Save**.
16. From the **Actions** menu select **Run...**



17. The Run OS Build Plan window is displayed. Confirm the Build Plan selected by pressing Next.



- 18. The next screen displays a list of available servers. Select the server on which to run the Build Plan (for this example, the previously network booted machine).
- 19. Press **Start**. The progress of the Run Build Plan job is displayed.

20. When the Run Build Plan job is complete, the server should be up-and-running and managed by SA.

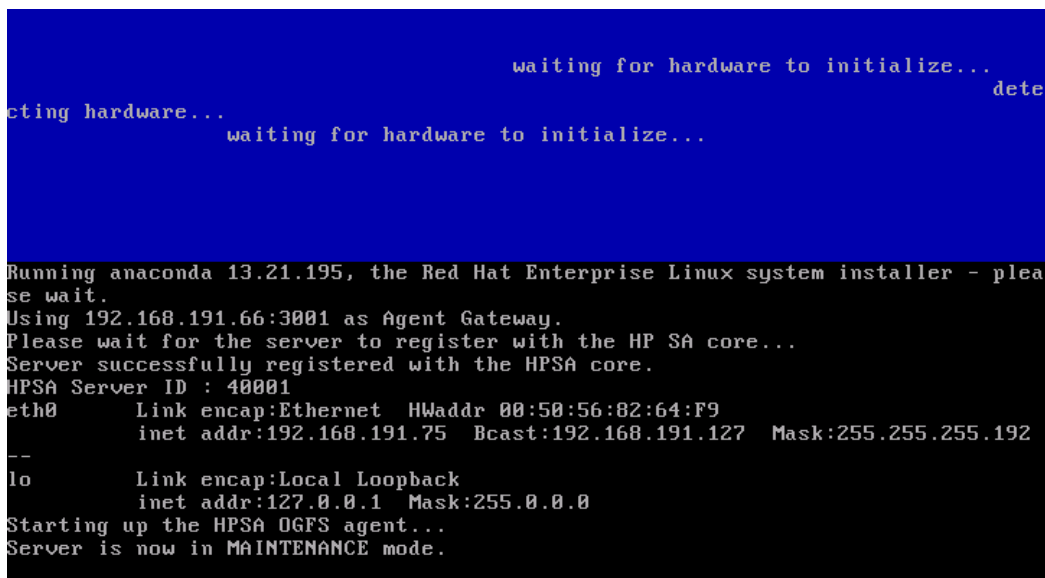
## Provisioning Linux-Based Servers

### *Provisioning Red Hat Enterprise Linux 6 x86\_64*

1. Network boot a server, as described in [Network Booting](#)
2. In the network boot menu, allow the `Autoboot:` option to boot (the default out-of-the-box is Linux 64-bit Service OS).

Alternatively, select **HPSA OS Build Plan Service OS Menu**, then choose Linux 64-bit Service OS.

Wait for the server to enter Maintenance mode.



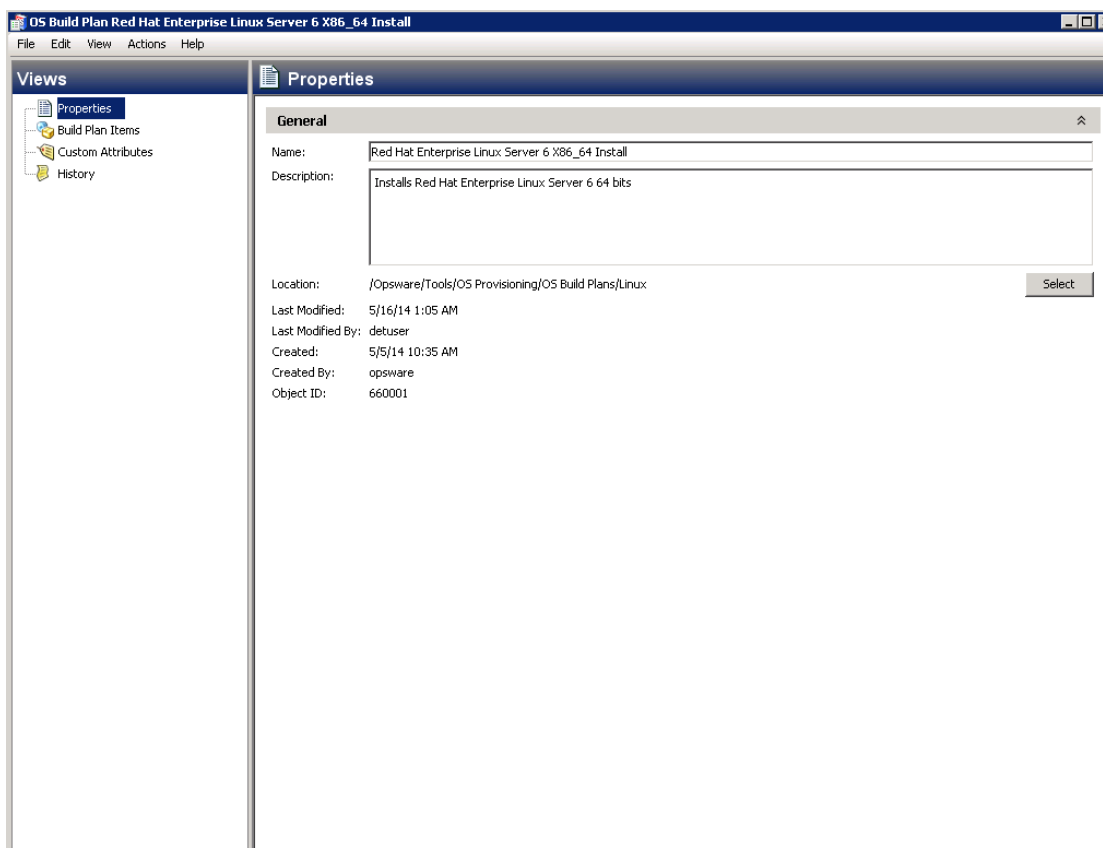
```

waiting for hardware to initialize...
cting hardware...
waiting for hardware to initialize...

Running anaconda 13.21.195, the Red Hat Enterprise Linux system installer - please wait.
Using 192.168.191.66:3001 as Agent Gateway.
Please wait for the server to register with the HP SA core...
Server successfully registered with the HPSA core.
HPSA Server ID : 40001
eth0      Link encap:Ethernet  HWaddr 00:50:56:82:64:F9
          inet addr:192.168.191.75  Bcast:192.168.191.127  Mask:255.255.255.192
--
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
Starting up the HPSA OGFS agent...
Server is now in MAINTENANCE mode.

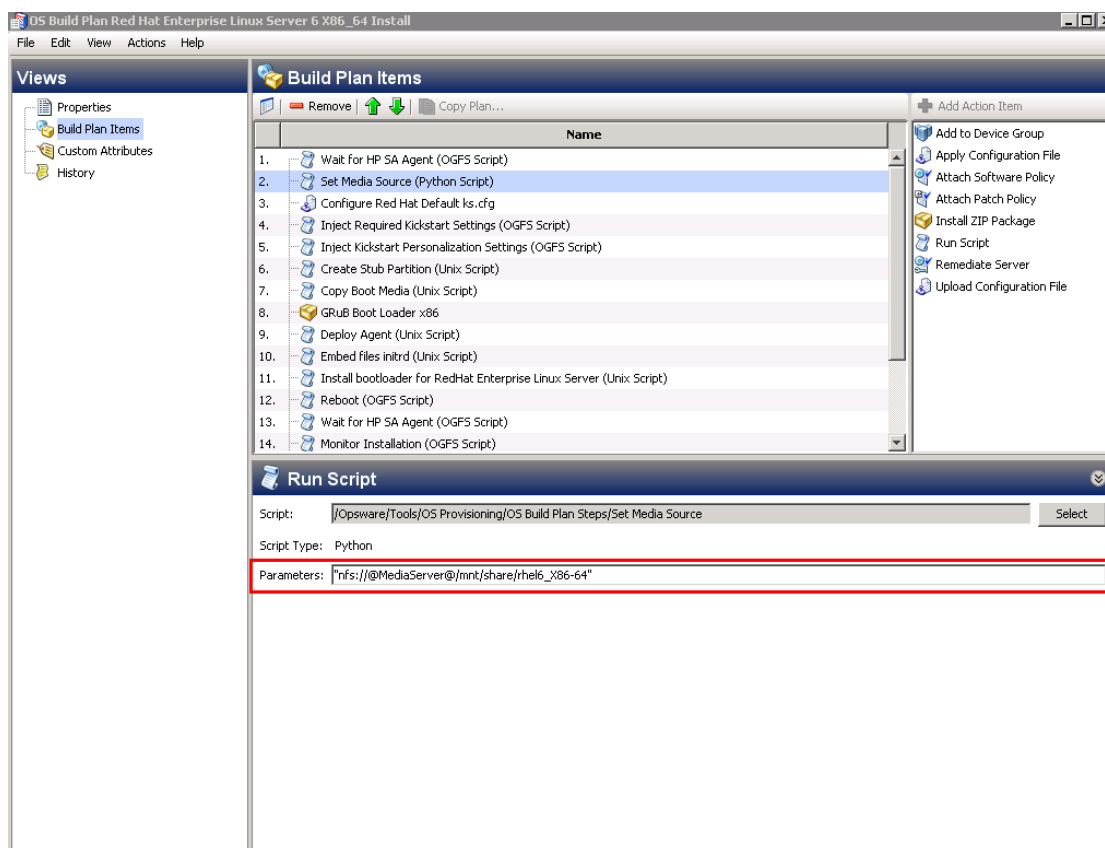
```

3. Note the server's IP address and the HPSA Server ID; these will help you identify the server later on.
4. Start the SA Client and in the Navigation pane, select Library and set the view to By Type and select the **OS Build Plans** folder.
5. From the displayed list of available Build Plans, select and open the Build Plan, Red Hat Enterprise Linux Server 6 X86\_64 Install.

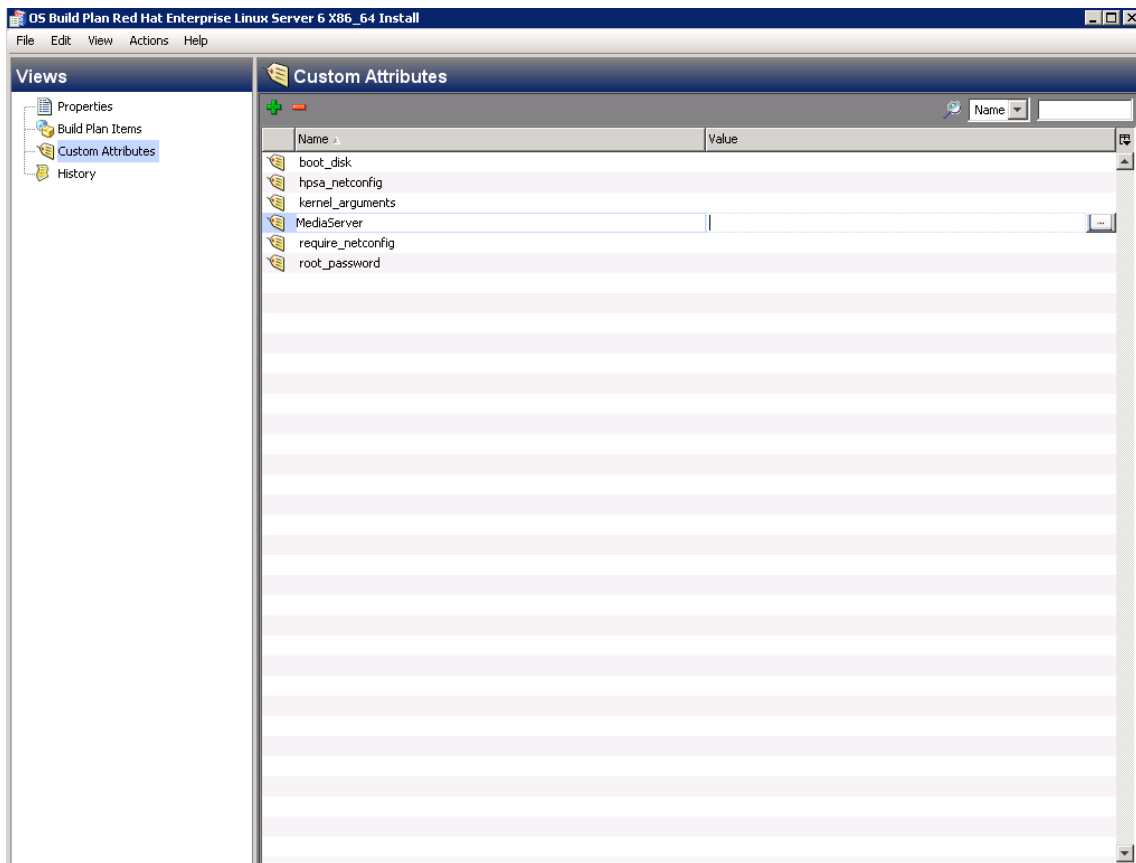


6. In the Build Plan Edit window, from the **Views** panel tree, select **Build Plan Items**. A list of Build Plan steps is displayed.
7. From the listed steps, select **Set Media Source**. The Run Script panel displays the Parameter setting for this step. Note the protocol, NFS, and the media path `/mnt/share/rhel_X86-64`.

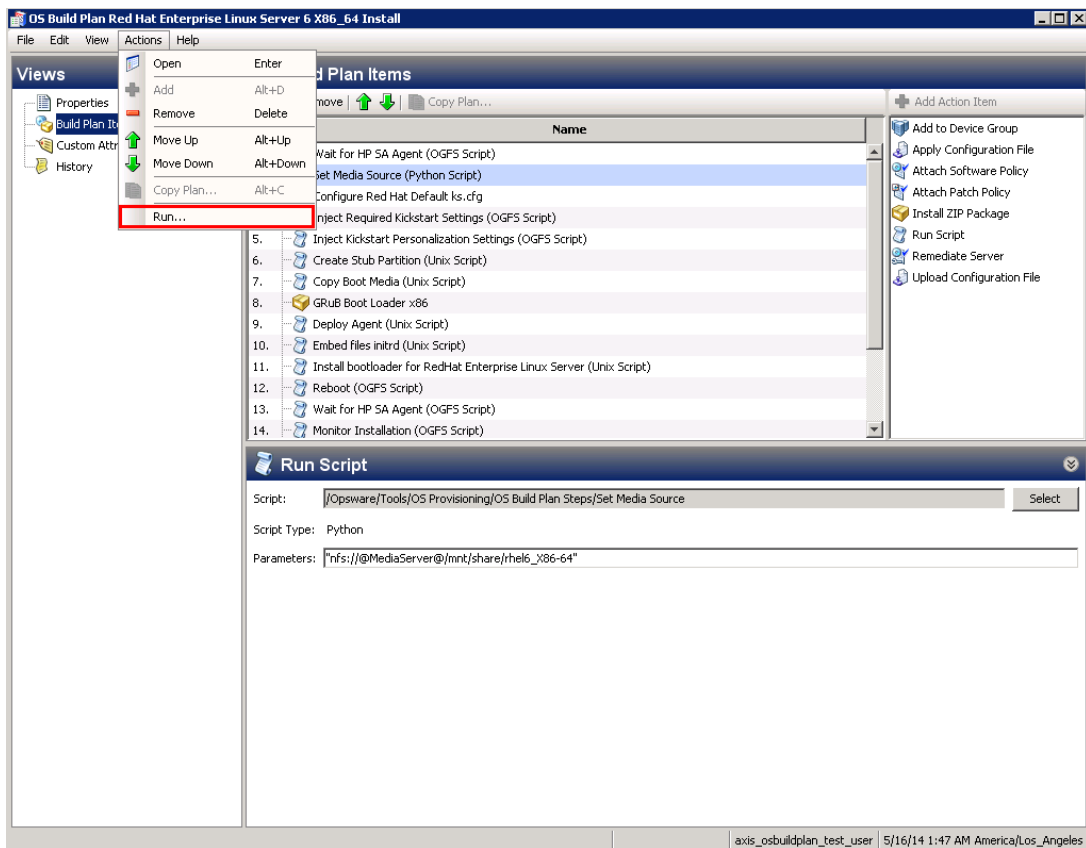




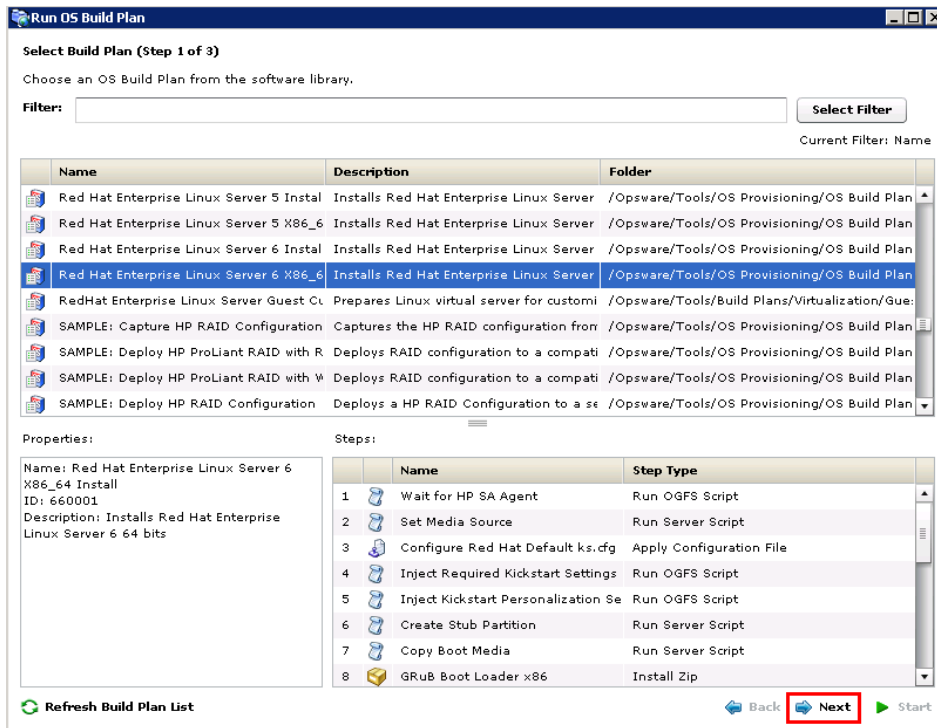
8. Prepare an NFS share with the Red Hat Enterprise Linux 6 x86\_64 DVD extracted under /mnt/share/rhel\_x86-64.
9. In the Edit Build Plan **Views** pane, select **Custom Attributes**.



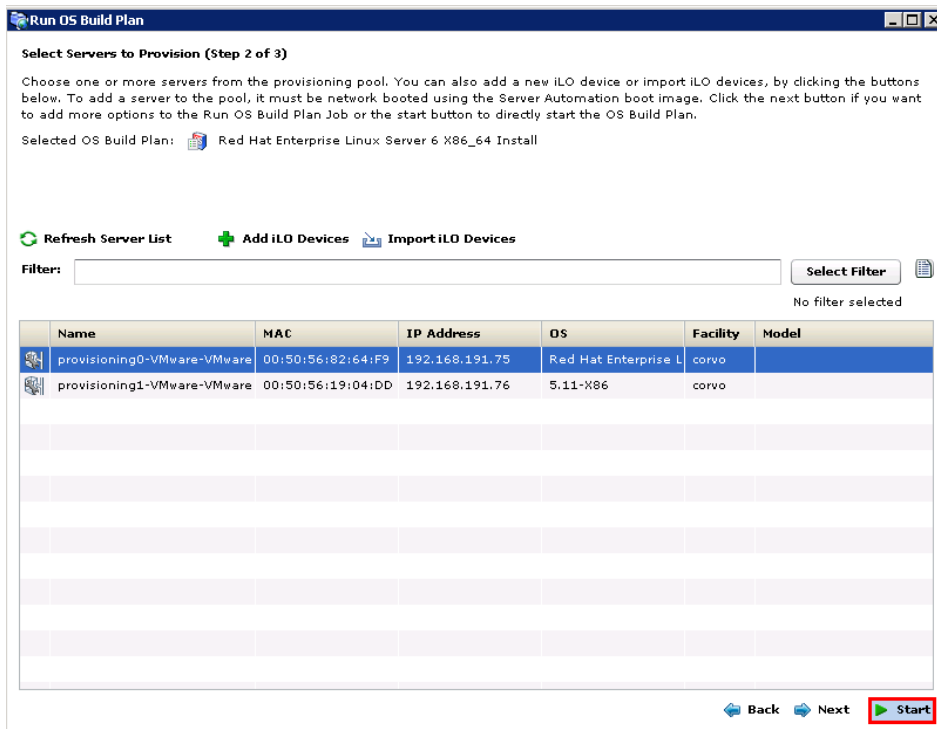
10. Complete the `MediaServer` custom attribute field with the hostname or IP address of the machine serving the NFS share you prepared previously.
11. Save the Build Plan by pressing **Ctrl+S** or use **File > Save**.
12. From the **Actions** menu select **Run...**



13. The Run OS Build Plan window is displayed. Confirm the Build Plan selected by pressing Next.



14. Select the server on which to run the Build Plan (the previously network booted machine).
15. Press **Start**. The progress of the Build Plan job is displayed.

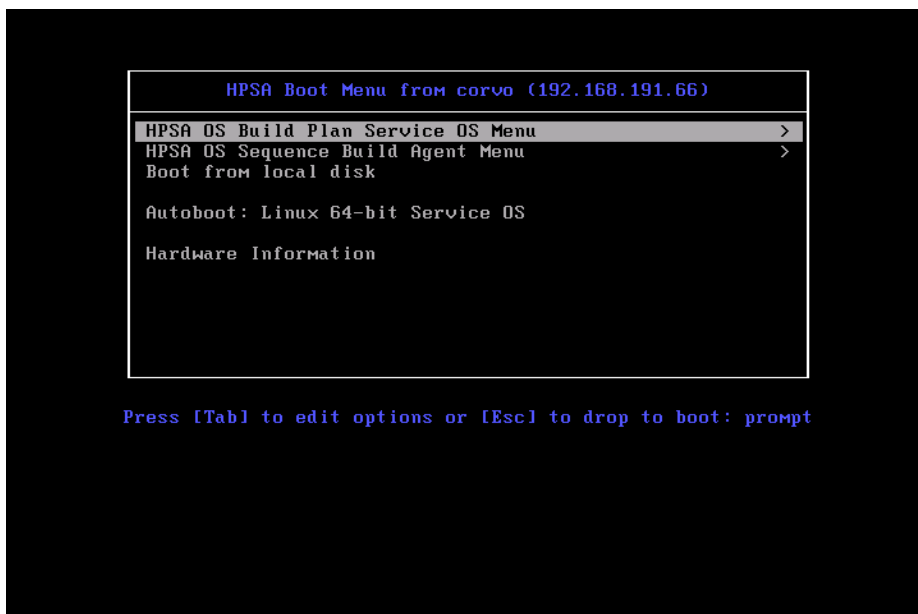


- When the Build Plan is complete, the server should be up-and-running and Managed by SA.

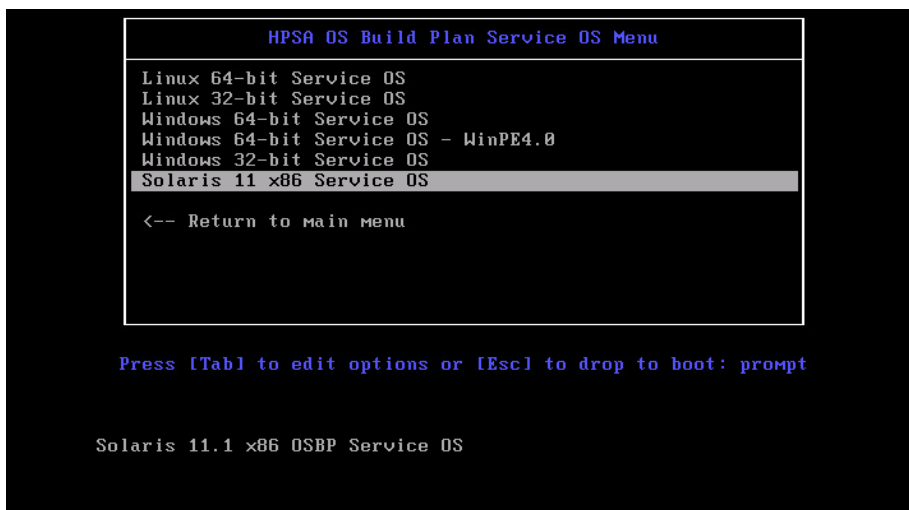
## Provisioning Solaris x86-Based Servers

### Provisioning Solaris 11.1 x86

- Network boot a server, as described in [Network Booting](#).
- On the network boot menu, select **HPSA OS Build Plan Service OS Menu**.



- Select Solaris 11 x86 Service OS.

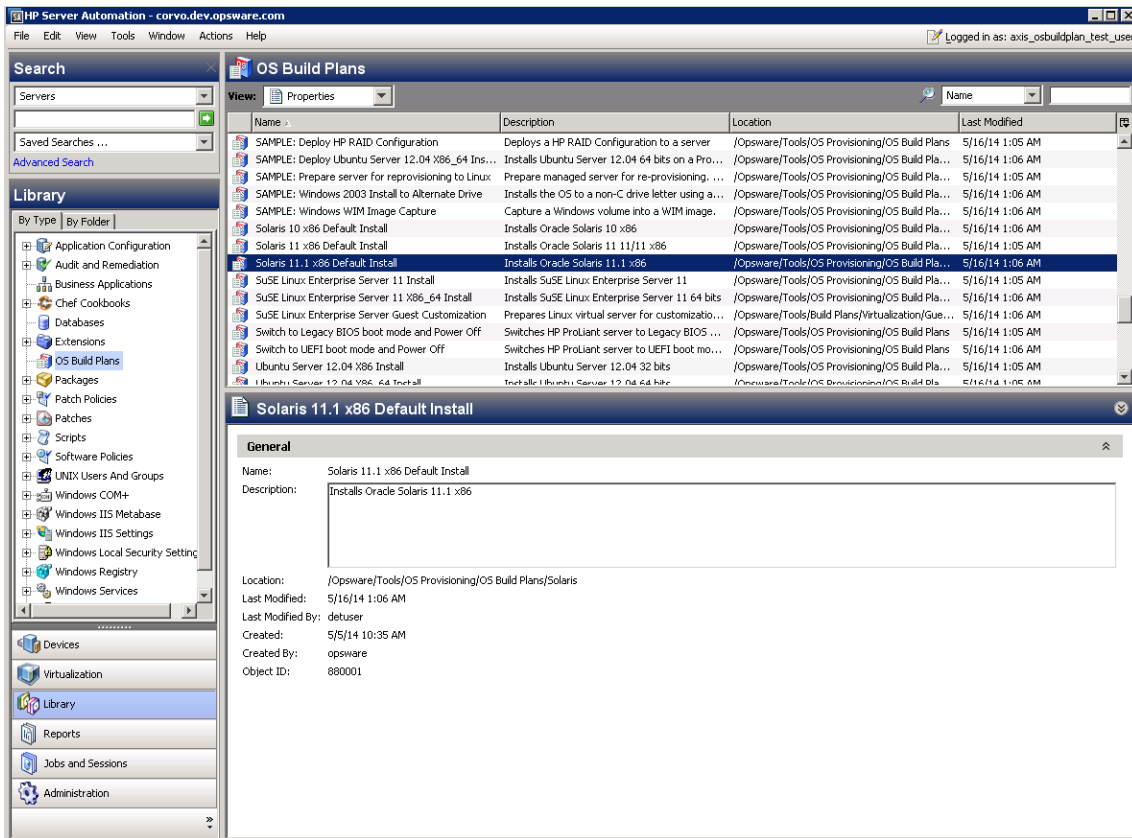


#### 4. Wait for the server to enter Maintenance mode.

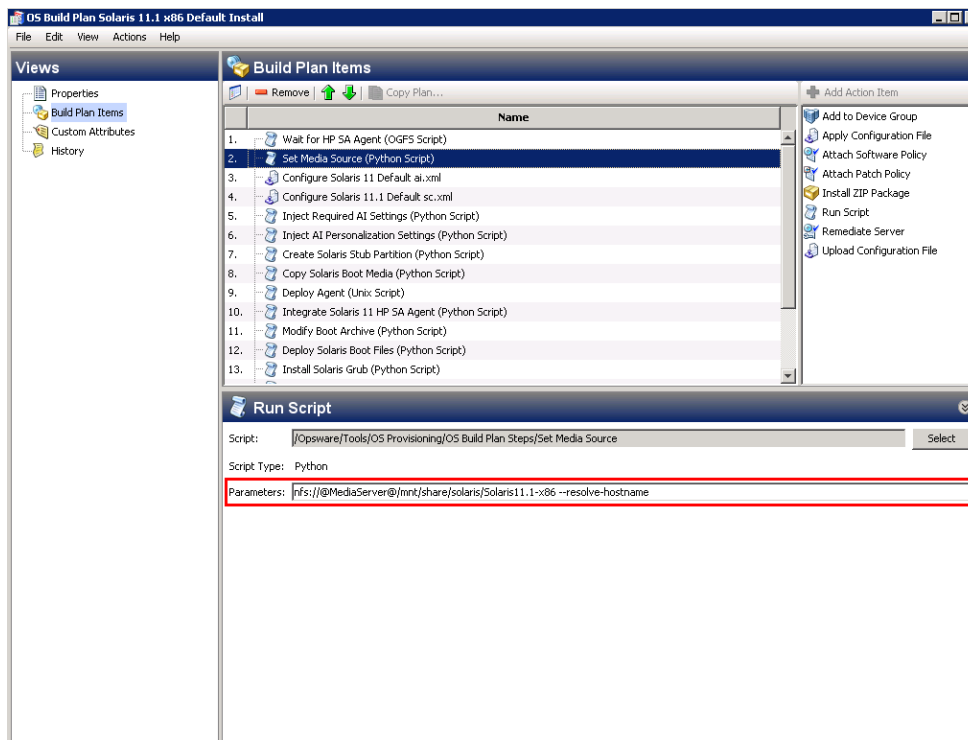
```
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
Downloading solarismisc.zlib
Downloading .image_info
Done mounting image
Configuring devices.
Hostname: provisioning4

provisioning4 console login: Using 192.168.191.66:3001 as Agent Gateway.
Please wait for the server to register with the HP SA core...
Server successfully registered with the HPSA core.
HPSA Server ID : 60001
Starting up the HPSA OGFS agent...
Server is now in MAINTENANCE mode.
█
```

5. Note the server's HPSA Server ID. This will help you identify the machine later on.
6. Start the SA Client and in the Navigation pane, select Library and set the view to By Type and select the **OS Build Plans** folder.
7. From the displayed list of available Build Plans, select and open the Build Plan, Solaris 11.1 x86 Default Install.



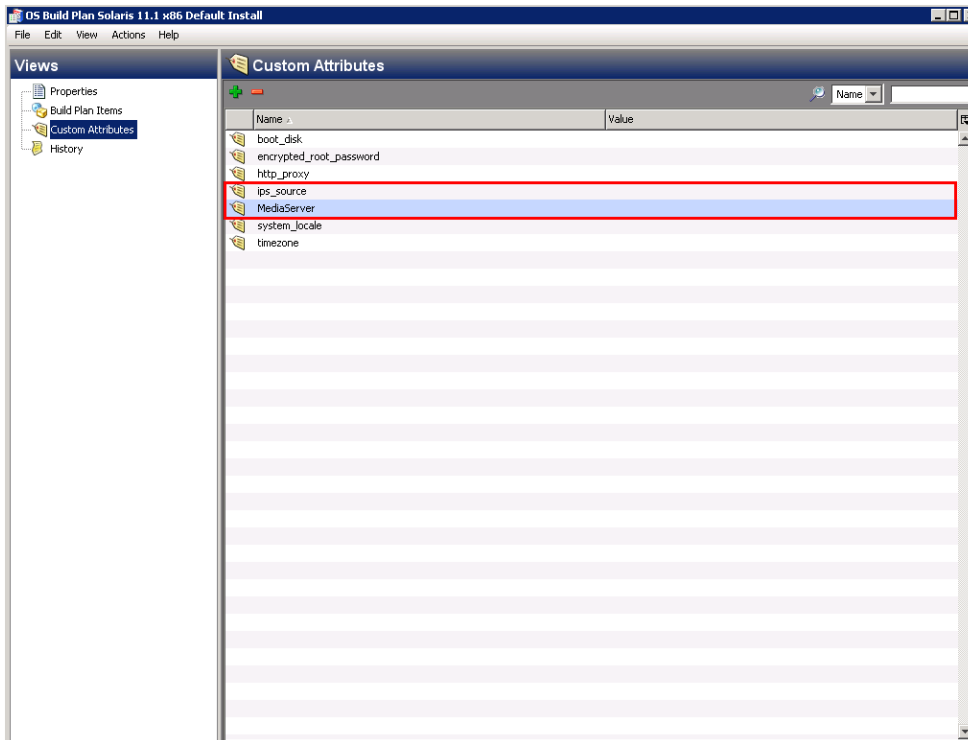
8. In the Build Plan Edit window, from the **Views** panel tree, select **Build Plan Items**. A list of Build Plan steps is displayed.
9. From the listed steps, select **Set Media Source**. The Run Script panel displays the Parameter setting for this step. Note the protocol, NFS, and the media path `/mnt/share/solaris/Solaris11.1-x86`.



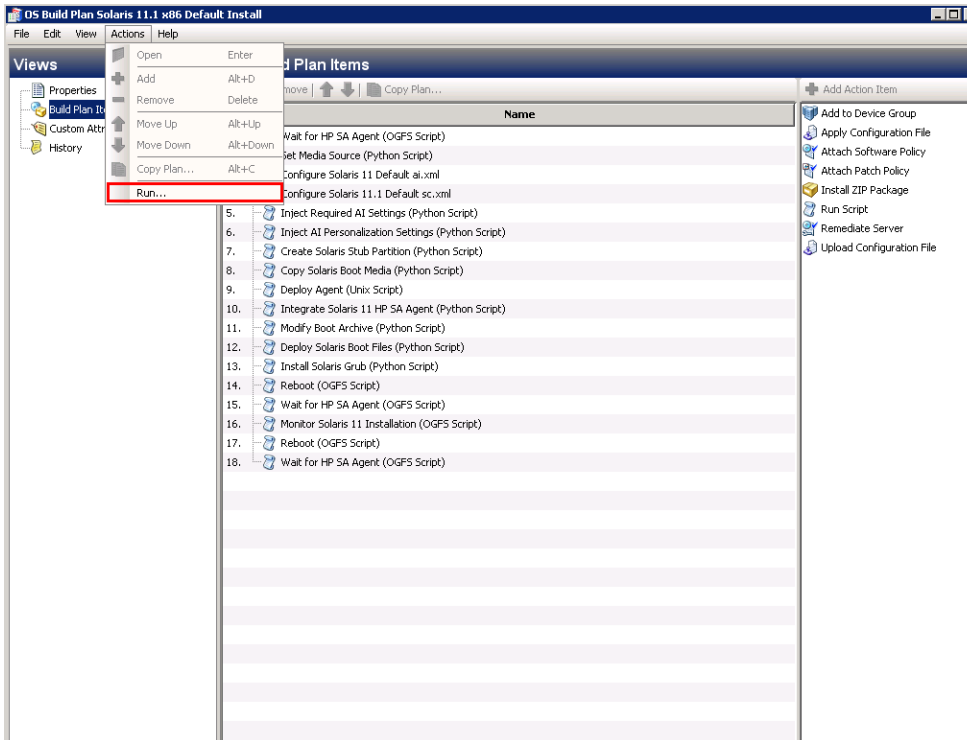
10. Prepare an NFS share using the Solaris 11.1 x86 AI Install CD extracted under
 

```
/mnt/share/solaris/Solaris11.1-x86
```
11. Ensure that your network has access to a Solaris 11.1 IPS package repository, either:
  - The official Oracle repository at <http://pkg.oracle.com/solaris/release/>.
  - A self-hosted repository available through HTTP (see your Oracle documentation for set up information)
12. In the Edit Build Plan **Views** pane, select **Custom Attributes**.

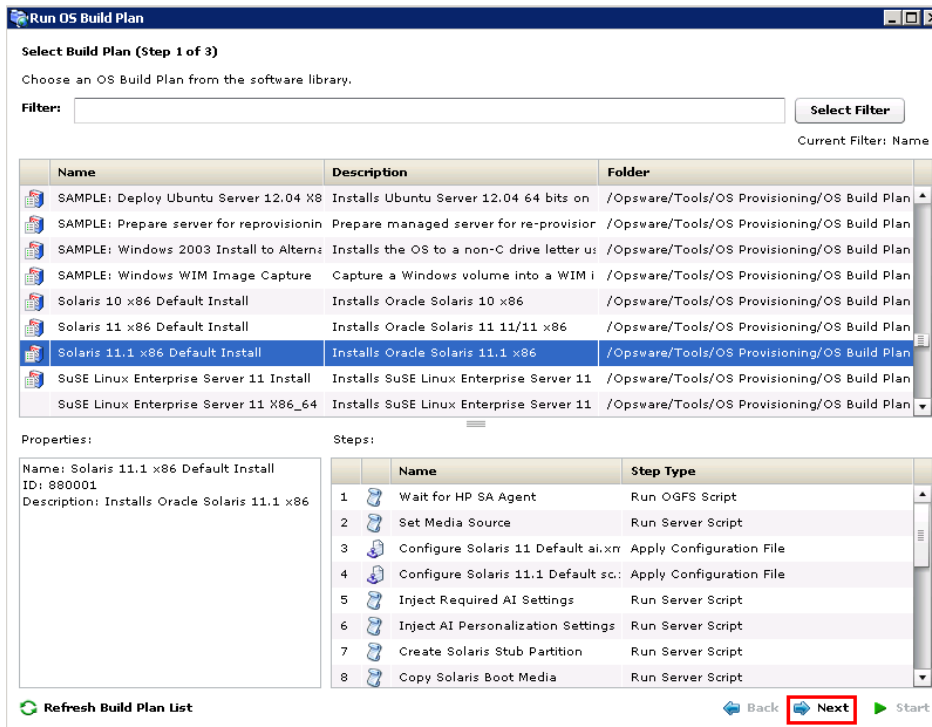




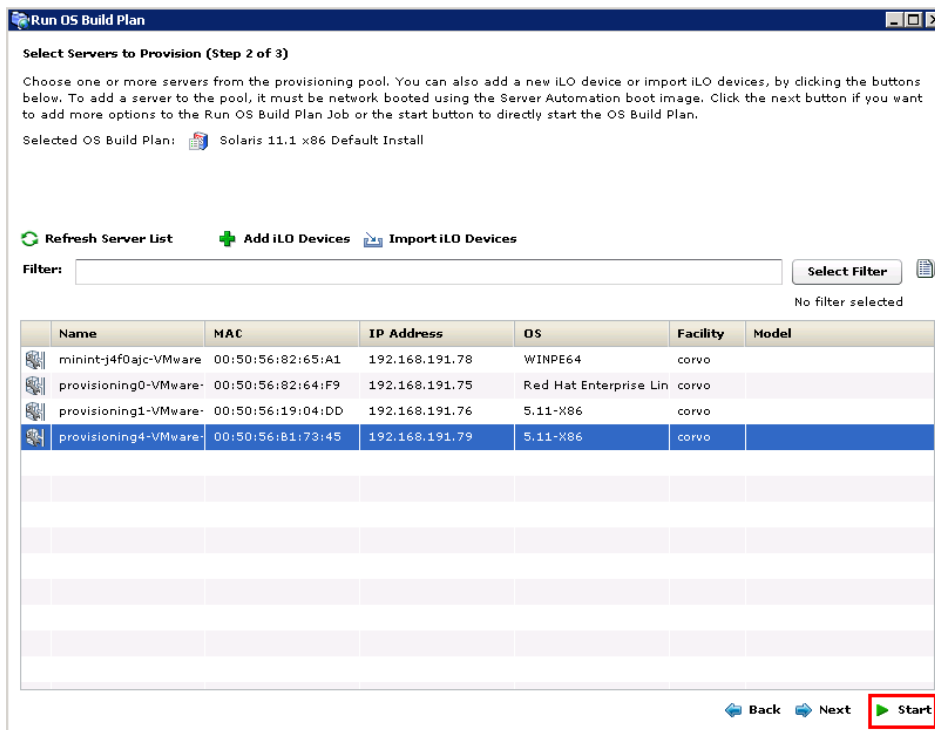
13. Complete the `MediaServer` and `ip_source` custom attribute field:
  - `MediaServer` must be the hostname or IP address of the NFS share you set up earlier.
  - `ips_source` must be the complete URL for accessing a Solaris 11.1 IPS package repository. **Optional:** specify the `http_proxy` custom attribute if the IPS repository is only accessible through a proxy.
14. Save the Build Plan by pressing **Ctrl+S** or using **File > Save**.
15. From the **Actions** menu select **Run...**



16. The Run OS Build Plan window is displayed. Confirm the Build Plan selected by pressing Next.



17. Select the server on which to run the Build Plan (the previously network booted machine).
18. Press **Start**. The progress of the Build Plan job is displayed.



When the Build Plan is complete, the server should be up and running, and Managed by SA.

## Provisioning ESXi-Based Servers

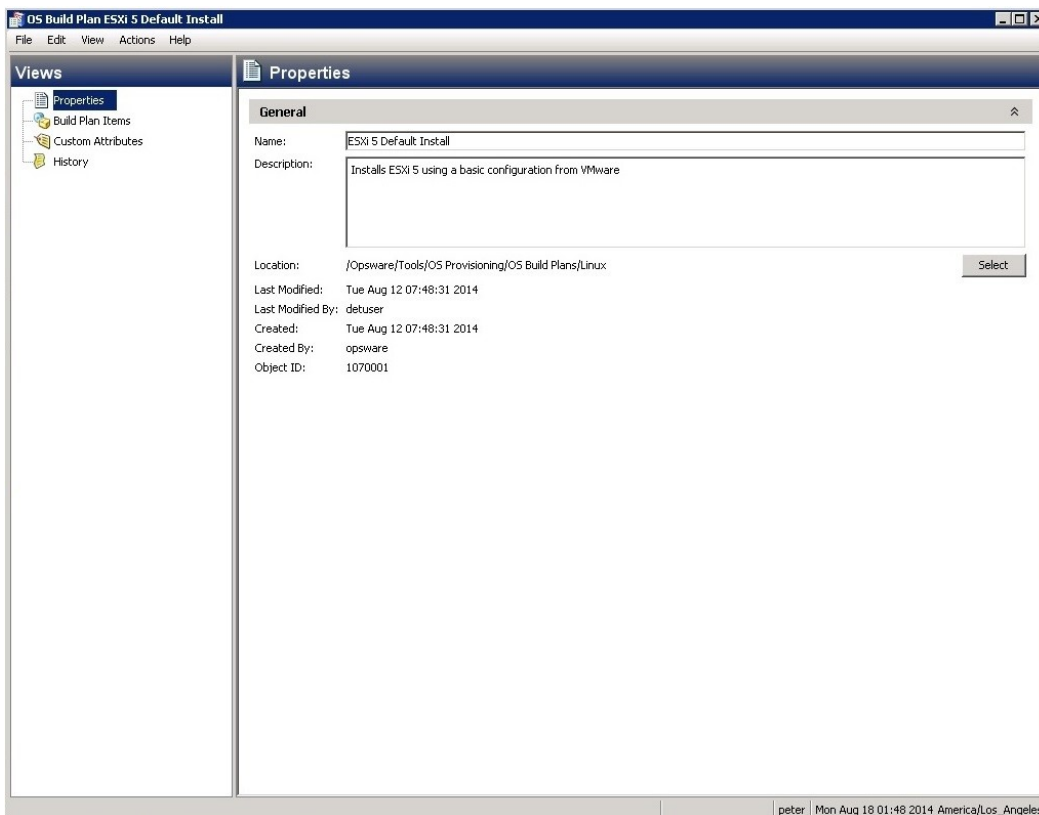
### Provisioning VMware ESXi 5.5

1. Network boot a server, as described in "Network Booting".
2. In the network boot menu, allow the `Autoboot:` option to boot (the default boot is to Linux 64-bit Service OS. Alternatively, select the **HPSA OS Build Plan Service OS Menu**, then choose Linux 64-bit Service OS.
3. Wait for the server to enter Maintenance.

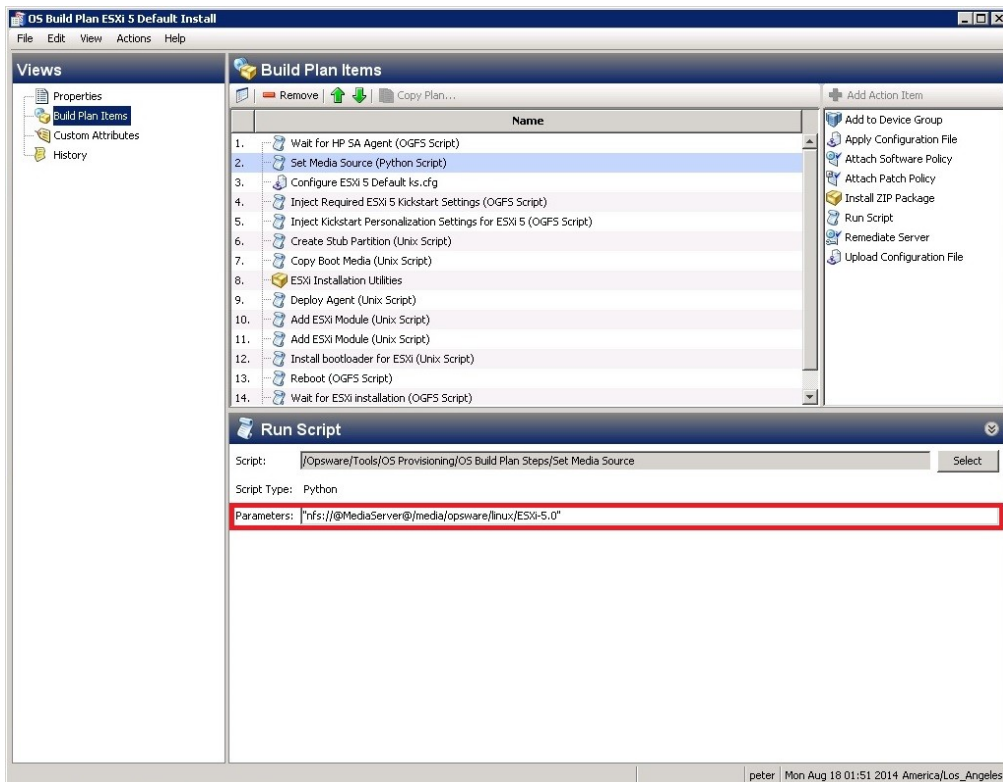
```
waiting for hardware to initialize...
detecting hardware...
waiting for hardware to initialize...

Running anaconda 13.21.195, the Red Hat Enterprise Linux system installer - please wait.
Using 192.168.59.2:3001 as Agent Gateway.
Please wait for the server to register with the HP SA core...
Server successfully registered with the HPSA core.
HPSA Server ID : 100001
eth0      Link encap:Ethernet  HWaddr 00:50:56:B1:02:23
          inet addr:192.168.59.163  Bcast:192.168.59.255  Mask:255.255.255.0
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
Starting up the HPSA OGFS agent...
Server is now in MAINTENANCE mode.
```

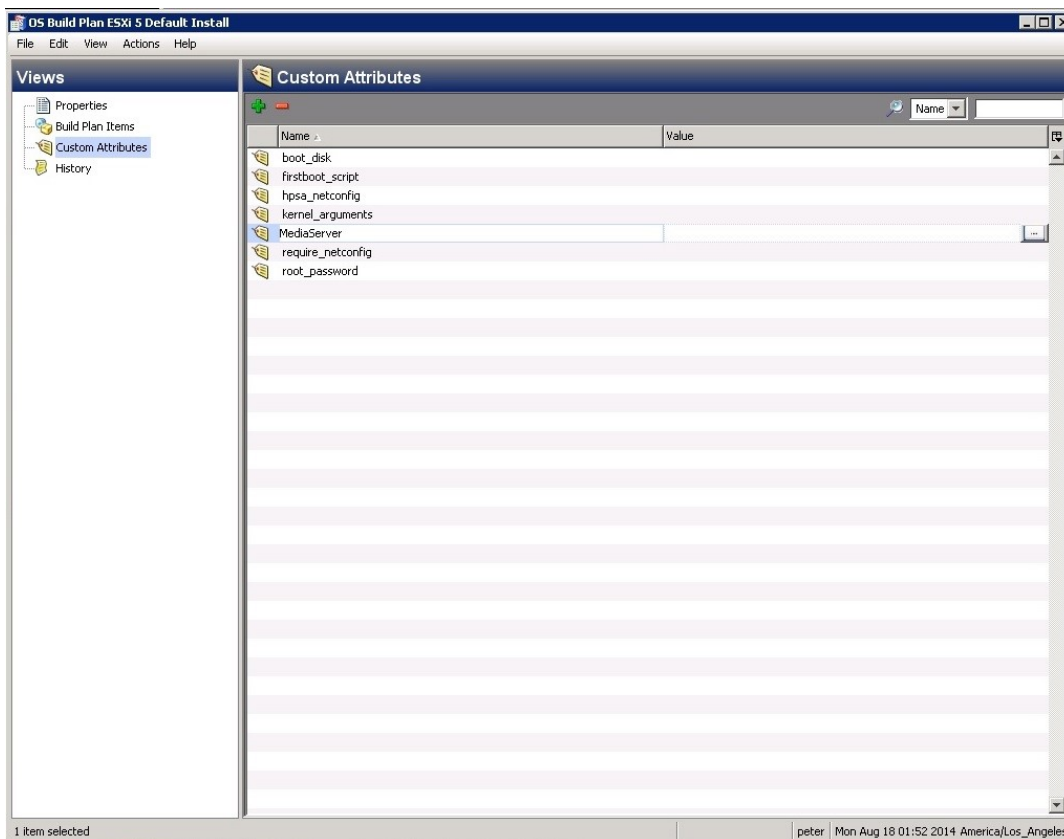
4. Note the server's IP address and the HPSA Server ID; these will help you identify the server later on.
5. Start the SA Client and in the Navigation pane, select Library and set the view to By Type and select the **OS Build Plans** folder.
6. From the displayed list of available Build Plans, select and open the Build Plan, ESXI 5 Default Install.



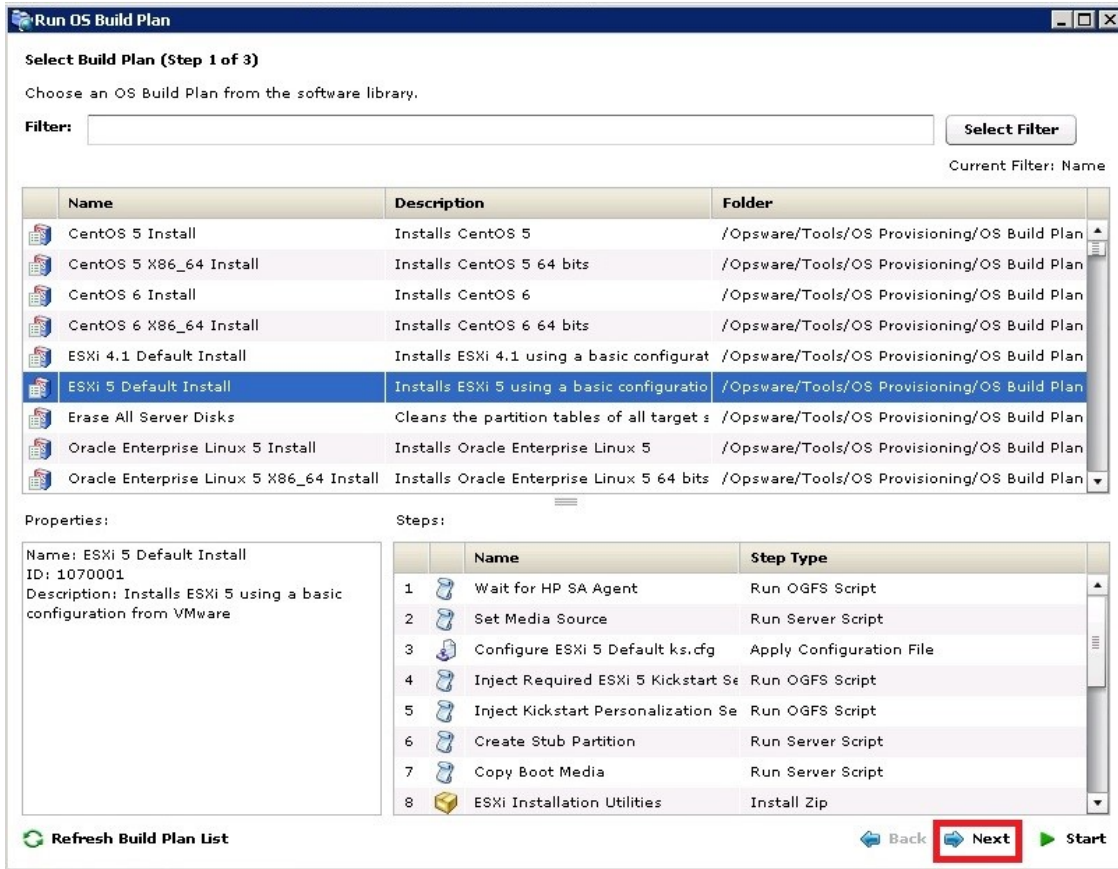
7. In the **Build Plan Edit** window, from the **Views** pane tree, select **Build Plan Items**. A list of Build Plan steps is displayed.
8. From the listed steps, select **Set Media Source**. The **Run Script** panel displays the Parameter setting for this step. Note the protocol, NFS, and the media path `/media/opsware/linux/ESXi-5.0`.



9. As described in the *User Guide: Provisioning*, "The Provisioning Process", "Phase1: Preparing the media", have a media share with the ESXi DVD extracted under `/media/opsware/linux/ESXi-5.0`.
10. In the Edit Build Plan **Views** pane, select **Custom Attributes**.

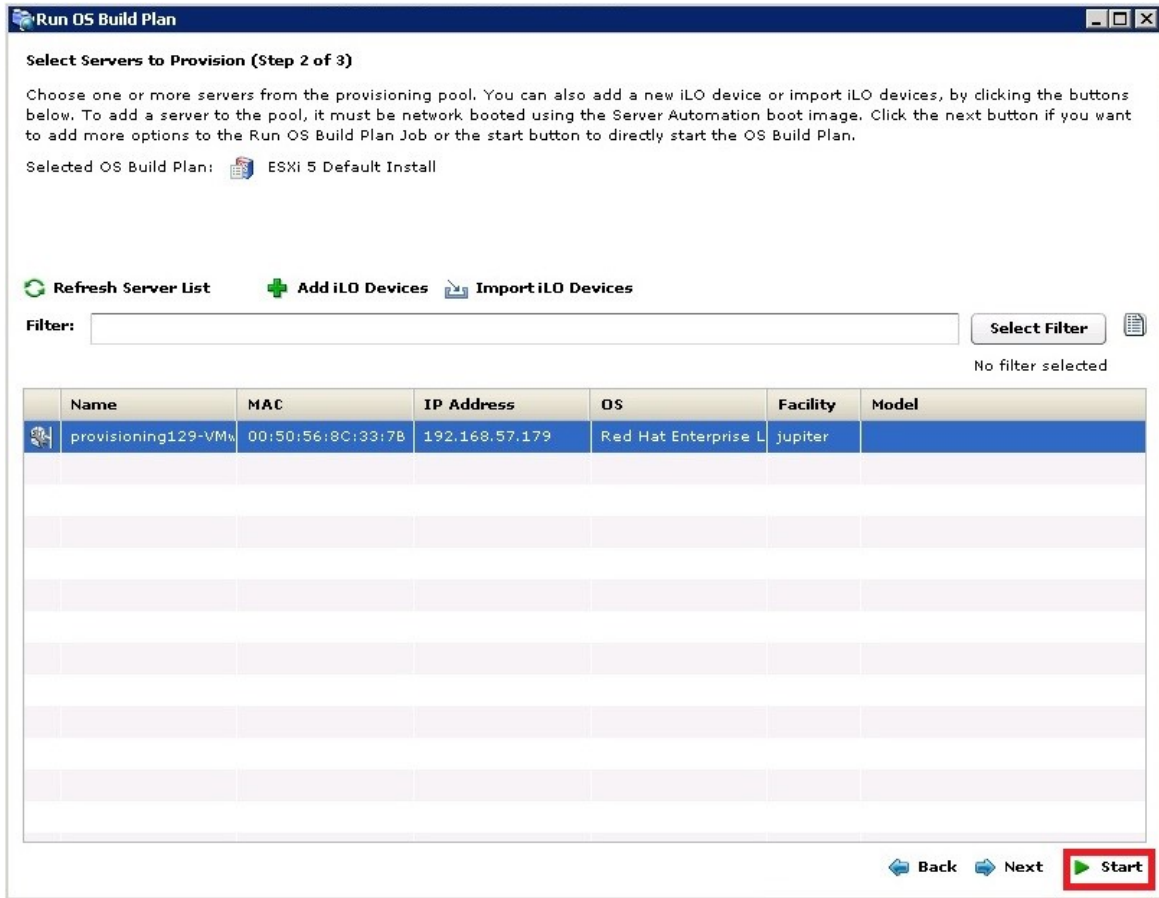


11. Complete the **MediaServer** custom attribute field with the hostname or IP address of the machine serving the share you prepared previously.
12. Save the Build Plan by pressing **Ctrl+S** or use **File > Save**.
13. From the **Actions** menu select **Run...**
14. The **Run OS Build Plan** window is displayed. Confirm the Build Plan selected by pressing **Next**.

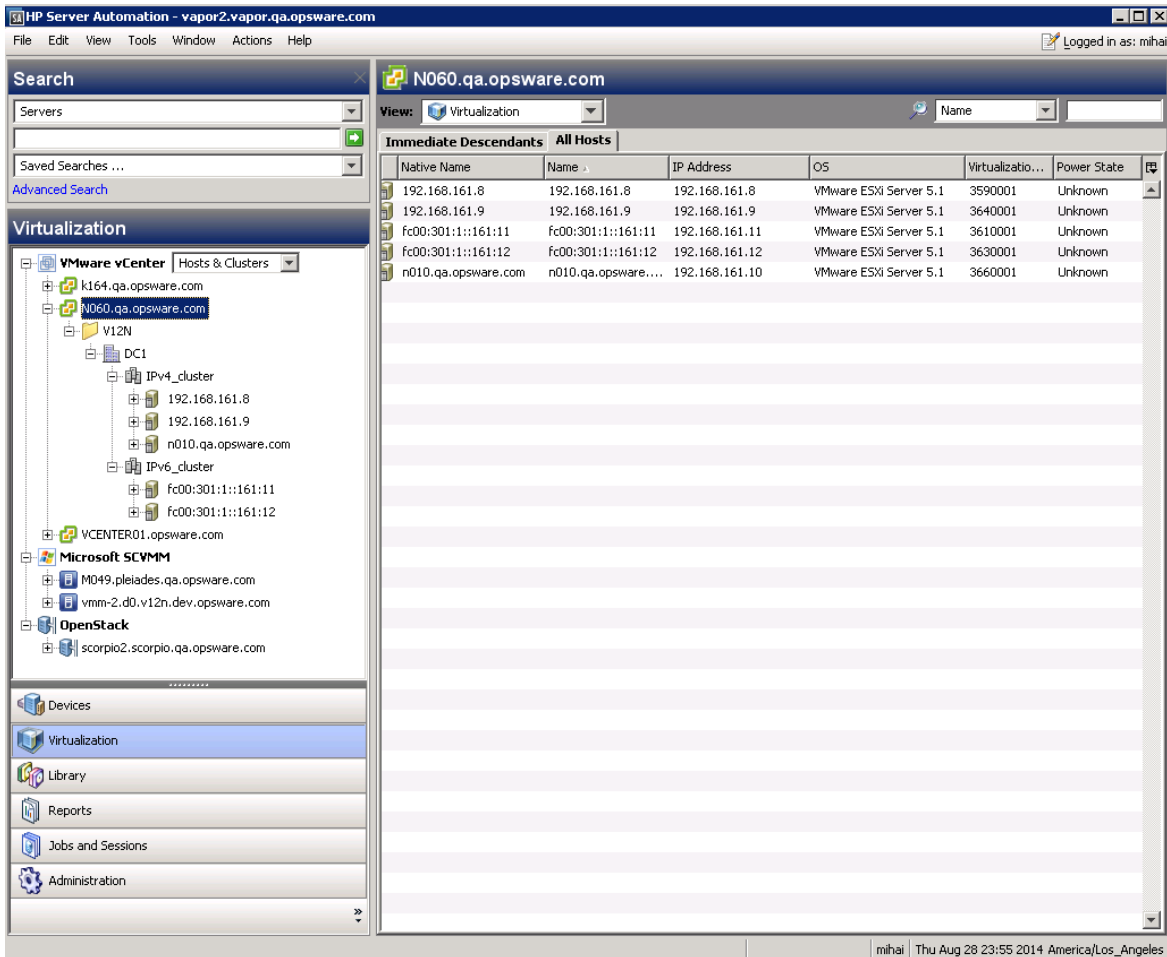


15. Select the server on which to run the Build Plan (the previously network booted machine).
16. Press **Start**. The progress of the Build Plan job is displayed.

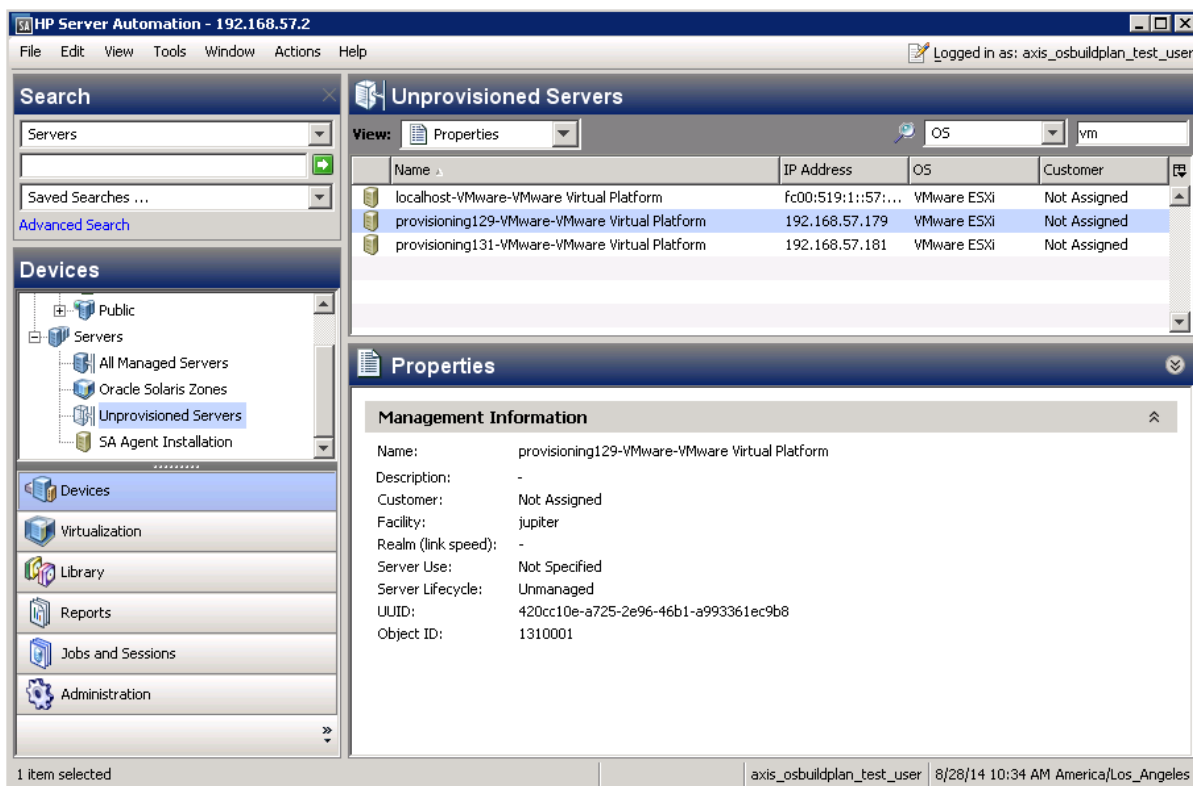




- 17. When the Build Plan is complete, the server should be up-and-running and managed by a VCenter Virtualization Manager if there is one registered to the SA Core. See also the *User Guide: Virtualization*.



**Note:** If there are no virtualization services registered, the Build Plan will be completed successfully and the server can be found under **Unprovisioned Servers** as an *unmanaged* server.



## Updating an HP ProLiant Server's Firmware

An HP Service Pack for ProLiant (SPP) is an ISO image that contains firmware, drivers, and software packages for Linux and Windows.

The components of each SPP are pretested together for stability.

To upgrade the firmware on target servers to the latest HP Service Pack, SA offers the baseline Build Plan `ProLiant SW - Offline Firmware Update`.

### Downloading an SPP

1. Go to <https://www.hp.com/go/spp>
2. Select **Download**.
3. Select **Current Version**.
4. Select **Download** next to the **Complete ISO Image** selection.
5. Enter your HP Passport user ID and password.
6. Follow the instructions to download the SPP ISO file.

### Deploying an SPP Image to Your Media Server

The HP-provided Build Plans that deploy SPPs require the SPP contents to be in the directory

`<File share name>/Media/spp`

on your Media Server.

Each SPP version has its own folder named after the SPP version.

For example, if the SPP is version 2014.06.0, the contents of the SPP ISO image should be extracted to

```
<File share name>/Media/spp/2014.06.0
```

on your Media Server.

### Preparing and Running the “ProLiant SW - Offline Firmware Update” Build Plan

**Note:** The offline firmware update must be performed in the Linux service OS. As such, the server will be automatically rebooted into the Linux service OS if needed.

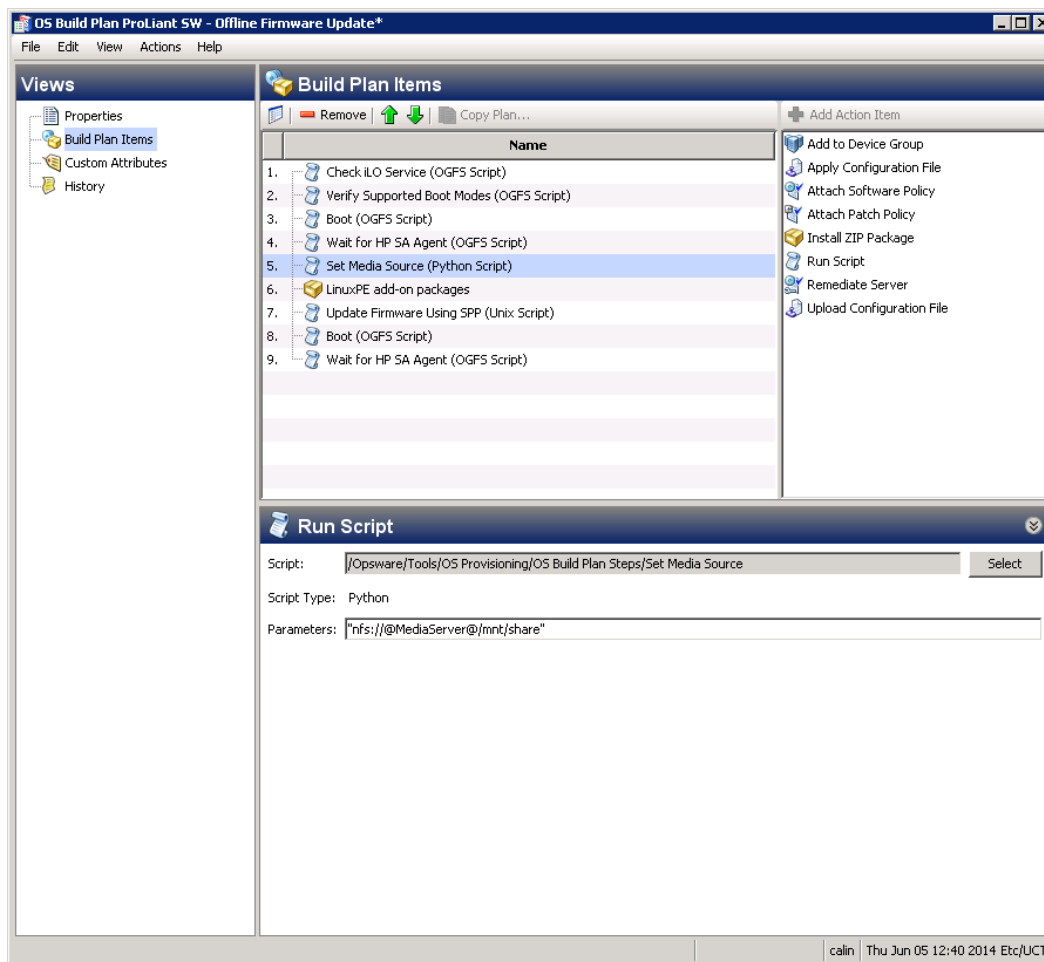
If you are updating firmware on a server already provisioned with an OS and you want your server booted back to this OS after the firmware update is complete, perform the following steps to create a Build Plan that boots your server back into its provisioned OS:

1. Copy the `ProLiant SW - Offline Firmware Update Build Plan` and save it under a new name.
2. Edit the Build Plan copy.
3. Delete the last two steps of the Build Plan: **Boot** and **Wait for HP SA Agent**.
4. Add a **Reboot** step at the end of the Build Plan.
5. Add a **Wait for HP SA Agent** step to the end of the Build Plan and specify these parameters:
  - `--production`
  - `--atLeast 3`
  - `--atMost 10`
6. Save the Build Plan.

The following steps must be completed before running the Build Plan:

1. Deploy the HP Service Pack for ProLiant (SPP) on your Media Server (see [Downloading an SPP](#) and [Deploying an SPP Image to Your Media Server](#)).
2. Edit the parameters of the `Set Media Source` step, specifying the media URI containing the SPP files (as specified in [Deploying an SPP Image to Your Media Server](#)).

Example, using NFS:



**Note:** When performing an offline firmware update, only the firmware components are installed from the SPP; software and driver components are not installed.

### Additional Parameters for the “Update Firmware Using SPP” Build Plan Step

- `--spp_version` - specify the SPP version to use. By default, the latest SPP version is used.
- `--hpsum_options` - specify any additional arguments to the HP SUM utility used to perform the firmware update.

For a detailed description of the available HP SUM options, refer to the `hp/swpackages/assets/doc/CLIHelp.txt` file in the SPP.

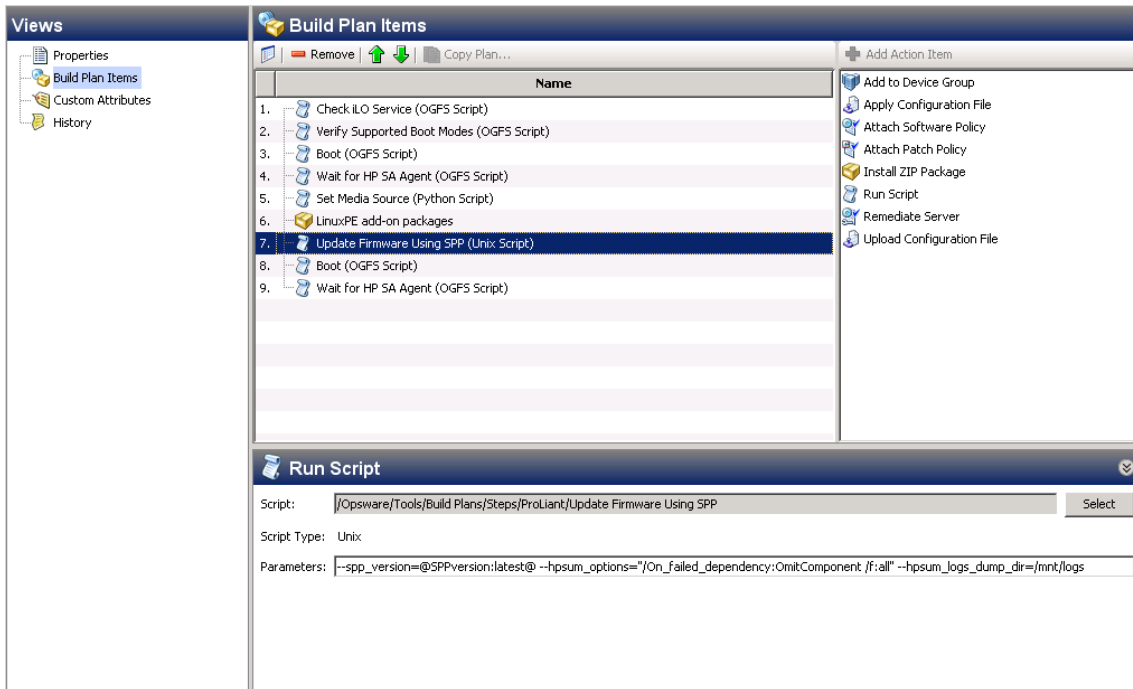
The syntax of this option is:

```
--hpsum_options="<option1 option2 ... optionN>"
```

- `--hpsum_logs_dump_dir` - specify a directory on a file share where a zip file containing the HP SUM logs will be written.

- `--no_show_log` - disables showing the `hpsum_log.txt` contents in the job log.

Example of a working parameter set:





# Creating New SA Build Plans

This section describes how to create new Build Plans, either by copying and modifying SA-supplied baseline Build Plans or manually creating new Build Plans.

**Caution:** Do not edit the source baseline Build Plans and Build Plan steps that are installed in the SA Client Library during SA installation. If you do so, when you upgrade SA any changes you made to these baseline Build Plans will be overwritten by the upgraded Build Plans. Always create renamed copies that you can then customize. These customizations will be preserved during upgrade.

**Note:** Making copies of Build Plan *script* steps that are installed by SA is not recommended, as you may miss out on updates and bug fixes.

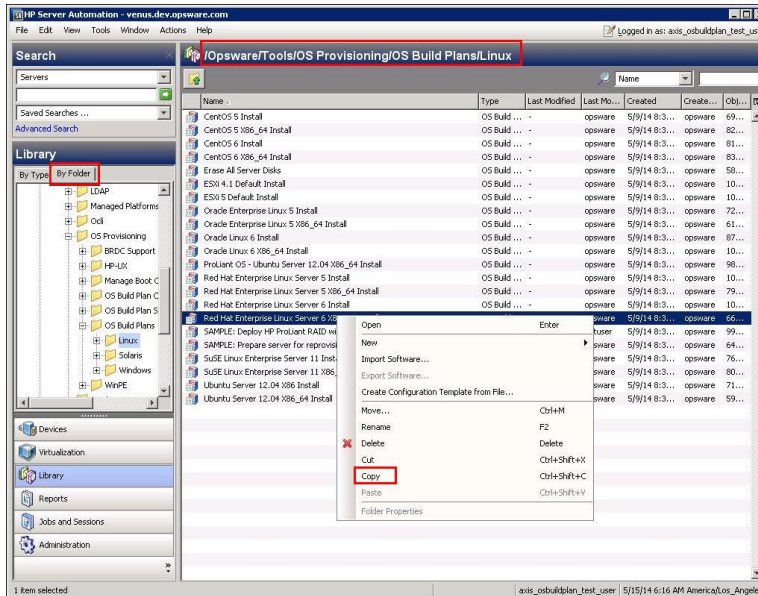
## Customizing Out-of-the-Box Build Plans

Before customizing a Build Plan, the Best Practice is to find a baseline Build Plan that is similar to the configuration you want to install and copy it.

To do this, perform these tasks

1. In the SA Client Library, select the **By Folder** tab.
2. Navigate to the folder containing the Build Plan you want to copy.
3. Select a Build Plan and select **Copy** from the **Actions** menu or from the context menu.





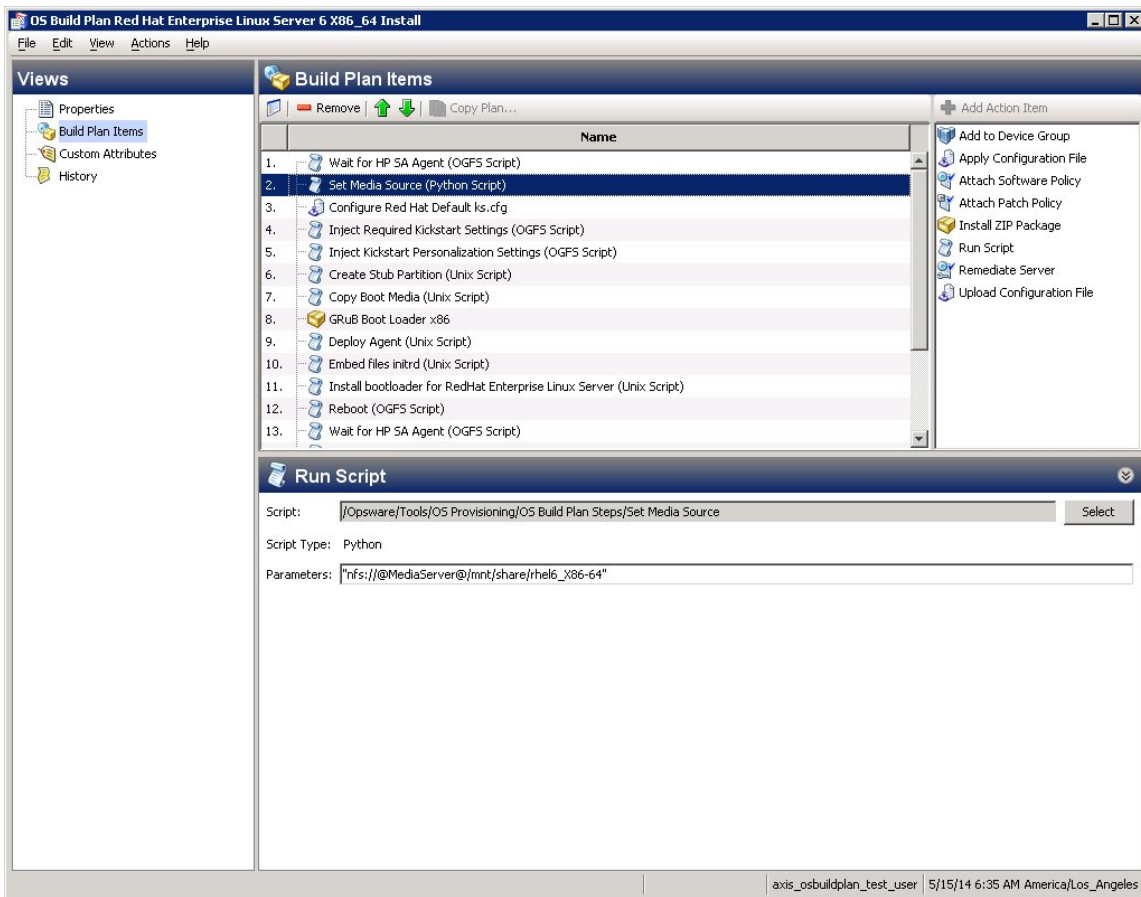
4. You can then paste the Build Plan to one of your own folders and open it to customize it. To avoid confusion, be sure to change the name and description.

You can also create a new Build Plan by starting the SA Client, selecting the SA Library panel in the Navigation pane, selecting the **By Type** tab and right clicking to select **New Build Plan** from the context menu.

You can also create a new Build Plan by starting the SA Client, selecting the SA Library panel in the Navigation pane, selecting the **By Type** tab and right clicking to select **New Build Plan** from the context menu.

### Editing a Build Plan

After opening a Build Plan, the **Edit Build Plan** window opens. You can add/edit steps and scripts, modify parameters for each step or delete steps and scripts.



**Note:** Be careful when changing the order of Build Plan steps (Items). Certain steps must be performed in a specific order and changing that order can cause errors.

**Note:** After upgrading to a newer release of SA, HP recommends that you compare the upgraded baseline Build Plans with any Build Plans you may have modified, since these might have improvements you could benefit from.

## Custom Attribute Substitution

The [Using Custom Attributes](#) section describes how to use the Build Plan step parameters using custom attributes for personalization.

Build Plans make use of custom attributes for unlimited personalization including the ability to define and use your own custom attributes.

As you can see in the Edit Build Plan Window above, Build Plans support a simple syntax for custom attribute substitution in the form:

```
@CustomAttributeName:default value@
```

This pattern is replaced by the custom attribute value you specify or the default value if there is no custom attribute specified. This substitution occurs in every text-based resource that Build Plans use as such as scripts, configuration templates and script parameters.

For example, if you want to specify a parameter for the media protocol for a Build Plan, edit the parameter of the **Set Media Source** step to:

```
@MediaProtocol:nfs@://@MediaServer@/mnt/share/rhel6_X86-64
```

This specification is persistent, unless you define a different `@MediaProtocol` custom attribute in the server hierarchy or in the Build Plan. The media protocol can be changed to, for example, HTTP by setting this Custom Attribute to HTTP.

Even though Build Plans are not attached to servers, custom attributes defined in the Build Plan are still used. The Build Plan custom attributes have a lesser priority and are overwritten if the same custom attribute is present on the server hierarchy. For more information about custom attributes and how they are inherited, see [Defining Custom Attributes](#).

## Customizing Installation Profiles

The installation profiles provided by SA are designed to install a minimal working system.

**Note:** These profiles should not be confused with the Installation Profiles used with the older SA OS Sequences.

You can use these profiles to:

- Customize partitioning
- Customize the firewall
- Install additional software from the installation media

**Note:** For ease of troubleshooting, you should not use any scripts as part of an installation profile. SA Scripts should be run as part of the Build Plan after the step that installs the OS. This makes the scripts easier to maintain.

For instructions about how to change individual installation profiles, see the vendor documentation for the OS you are installing and for the supported syntax. The name of the configuration template provides keyword suggestions searching the syntax.

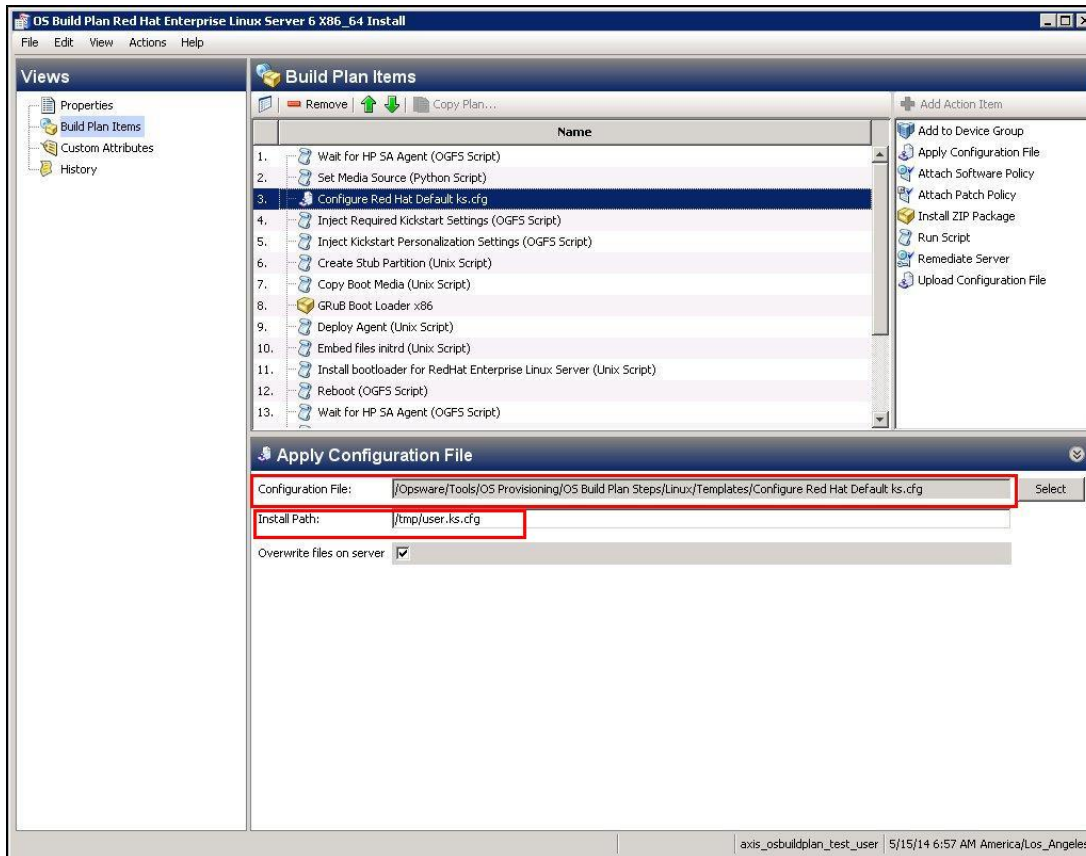
Depending on the OS that you are installing and how much you want to change, you can either create a new profile or copy and modify an existing profile.

### Modifying an Existing Installation Profile

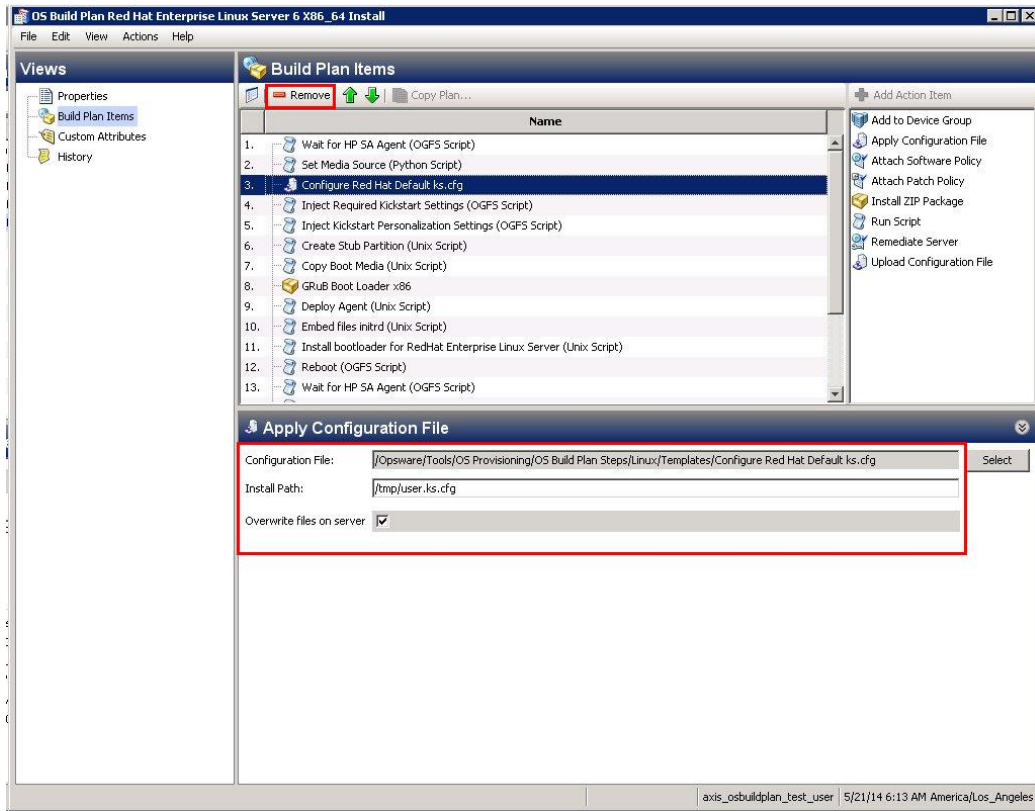
To modify an existing installation profile that is already in the Build Plan:

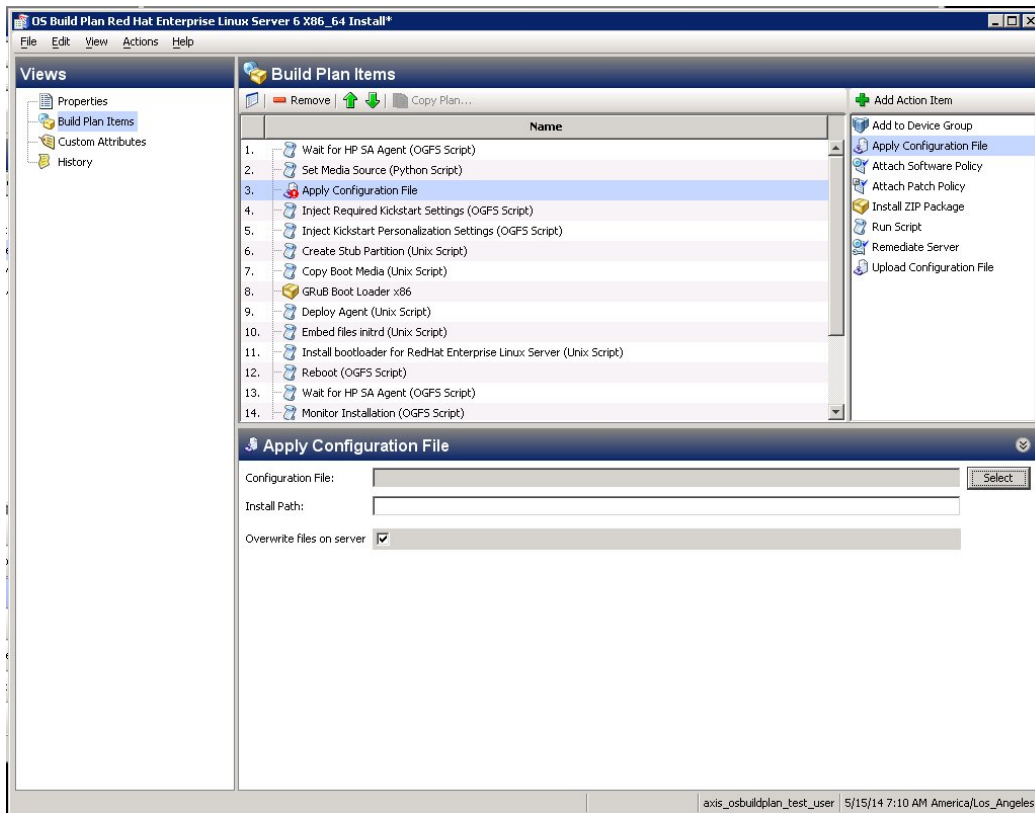
1. Start the SA Client and in the Navigation pane, set the view to the **By Type** tab.
2. Select the folder **OS Build Plans**.
3. From the list of available plans, open the plan you intend to modify.

4. Select the step in the Edit Build Plan window and note the location as shown below:



5. Select the **Apply Configuration File** step. Note the **Install Path** and the step's order position in the **Build Plan**. You will need this information when adding your own **Apply Configuration File** step.
6. In the SA Client Library panel, select the **By Folder** tab, locate the configuration file and create a copy of the file.
7. Open the configuration file copy and in the edit window, remove the old step in the **Build Plan** by clicking on the **Remove** button.
8. Add the new step as shown below:





Note that the **Apply Configuration File** step applies only to the configurations from Configuration Templates.

9. Ensure that the **Apply Configuration File** step is in the correct order and has the same `Install Path`.
10. After you add the **Apply Configuration File** step, you can double click it to edit it if necessary.

You can also create a new Configuration File in the SA Client Library **By Folder** view by selecting the **New...** context menu option and selecting **Configuration Template**, or by navigating to the **Configuration Templates** folder when viewing the SA Client Library in **By Type** view. Be sure to change the Parser Syntax. Build Plans only support the *Custom Attribute Syntax*. You can copy/paste your installation profile in the content section of the Configuration Template. You can add the new Configuration Template in the same way as before.

Build Plans modify the installation profile using the `Inject Required` and `Inject Personalization` family of scripts that are specific to each profile type. The `Inject Required` scripts change the profile to integrate it with the SA Provisioning process. For example, the path to the install media is added to the profile. The `Inject Personalization` scripts extend functionality in addition to the profile. For example, steps that add network configurations to the profile.

## Network Setup

Build Plans support customizing the network settings in all stages of the install process including the final OS. For this reason, network settings present in the Build Plan take precedence over those settings specified in the installation profile.

Network settings in the installation profiles are optional. For more information on how to configure the network with SA Provisioning, see [Personalize Network Settings](#).

## Firewall Considerations

SA Provisioning Build Plans may make minor modifications to the firewall configurations on managed servers on Windows platforms so that communication between the SA Core and the SA Agent is not blocked.

For other managed server platforms, the installation profiles may disable the firewall.

**Note:** If you customize installation profiles to contain a firewall, ensure that the connection between the SA Core and the SA Agent is not blocked.

## Examples

For Red Hat Enterprise Linux, the following line in your `ks.cfg` profile enables the firewall and allows the SA Agent to function correctly:

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

For SUSE Linux Enterprise Server, the following lines in the `autoyast.xml` profile enable the firewall and allow the SA Agent to function correctly.

```
<firewall>
<FW_SERVICES_EXT_TCP>1002</FW_SERVICES_EXT_TCP>
<enable_firewall config:type="boolean">true</enable_firewall>
<start_firewall config:type="boolean">true</start_firewall>
</firewall>
```

## Flow Control Mechanism

The *Flow Control Mechanism* is used to control the flow of the Build Plan in a linear way, for example you can use it to restart a Build Plan from a step that failed in a previous job or to skip a few steps in a special case

A Flow Control step is an OGFS or Server script that prints specific key words (grouped as a section) that are interpreted by the Build Plan Runner as a Flow Control instruction. For example:

```
Begin Flow Control:
Forward to step 16
End Flow Control
```

The **Forward to step value** instruction, instructs the Build Plan runner to continue with a forward step (it cannot go backwards or return to the same step). Accepted values for this instruction:

<b>X</b>	Where X is a positive integer, will make the Build Plan runner to continue with Build Plan step X - <b>absolute step number</b>
<b>+X</b>	Where X is again a positive integer. <b>Default:</b> no steps are skipped, This value makes the Build Plan Runner jump forward relative to the current step - <b>relative step number</b> . For example, the value +1 causes a jump to the next step. The value +2 causes the next step to be skipped (jump to the second step following the current step).
<b>\$END</b>	Causes the Build Plan to finish execution.

**Note:** You cannot have negative integer values or strings other than \$END.

If you specify an invalid Build Plan step, the Build Plan ignores the instruction and finishes execution.

The Flow Control section once found is removed from the script output so it is not be printed in the Build Plan output.

To help with the implementation of Flow Control Scripts, both for OGFS and Server scripts, use the following environment variables

<b>BUILD_PLAN_ID</b>	the current Build Plan ID
<b>TARGET_SERVER_ID</b>	the current target server ID
<b>CURRENT_STEP_NO</b>	the currently running Build Plan step number

SA provides a sample step and a sample Build Plan that you can use to understand how flow control works.

- The sample `Continue with previous job run failed step` must be the first step of a Build Plan. It generates a flow control statement that causes the Build Plan to forward to the step where it last failed on the same server. The step has no effect if the Build Plan has not been run previously ran or if it succeeded.

**NOTE:** The step does not take into consideration any Build Plan changes, so if the steps changed since the last run, you could get unexpected results, as the step will forward to a step that may be different than the one at which the Build Plan previously failed.

- The Build Plan `SAMPLE: Flow Control Restart from failed step` is an example of how **Continue with previous job run failed step** should be used. Rerun the sample Build Plan until it finishes successfully to see what the first step does.



## Build Plan Steps

The following section describes SA Build Plan steps.

### Run Script Step

The `Run Script` step is a key Build Plan component. The vast majority of steps used in Build Plans make use of this step. The `Run Script` step executes a script either on the target server or in the SA Global Shell (OGFS). SA Provisioning provides an extensive library of scripts that perform many of the most common Build Plan tasks. In addition, you can create your own scripts by copying and modifying the scripts that HP provides or by creating new scripts.

The following are some of the script types SA provides:

**Table: Build Plan Scripts**

Script Type	Description
<b>OGFS scripts</b>	Execute on the SA Core in the SA Global Shell. All other script steps run on the target server.  <b>Note:</b> Most of the OGFS scripts shipped with SA are not meant to be modified in any way. They provide vital functionality like booting target servers and monitoring tasks.
<b>Python scripts</b>	Execute on the target server. This is the only script step that can be run against any target server leveraging the platform independence of Python.
<b>Standard UNIX/Linux shell scripts</b>	Execute on the target server. These scripts make use of any interpreter that is installed on the server.
<b>Standard Windows batch scripts</b>	Execute on the target server.
<b>Standard Windows Visual Basic scripts</b>	Execute on the target server.
<b>PowerShell</b>	PowerShell script steps are not currently supported. As a workaround you can create a Windows batch script that calls the PowerShell interpreter.

OGFS scripts are started at the following location:

```
/opsw/.Server.Id/<<target server ID>>
```

From the Global Shell you can use the UAPI as the user under which the Build Plan runs. For more information about the UAPI and the Global Shell see the *SA User Guide: Server Automation, "SA Global Shell"*. Server scripts can also access the UAPI but will do so using the credentials of the target server, thus limiting operations to that server.

## Install Zip Step

The Install Zip step transfers a zip file from the SA Client Library to a target server and uncompresses it to a specified location. You can also specify pre-installation and post-installation scripts with the package which can execute before or after file extraction.

**Note:** Zip packages installed using Build Plans are not added to the server's software inventory. If you want these packages to show up in the inventory, add the Zip to a Software Policy and use the **Attach Software Policy** and the **Remedate Server** step (applicable to servers running the production SA Agent). There is no equivalent for servers running the service OS as the service OS is not persisted and the contents of the installed Zip is lost when the server is rebooted.

## Capture and Deploy Configuration File Steps

Configuration text files are stored in the SA Client Library and are used for text-based data such as unattended installation files or hardware configuration files. The **Deploy Configuration File** step takes the specified configuration file and writes it to a user-specified location on the target server. Such steps are often followed by a run script step that makes use of the configuration file. HP provides many sample configurations. You can use these configurations or create your own.

The **Capture Configuration Files** step captures a configuration text file from the target server and uploads it to the SA Client Library so that it can later be used as part of a **Deploy Configuration File** step.

See [Add to Device Group Step](#) and [Attach Software or Patch Policy Steps](#).

## Add to Device Group Step

This script causes a server to join a static device group. This script is typically used to configure custom attributes on the device group and have them inherited by the server so they can be used during the provisioning process. Custom attributes defined on the device group will be accessible after the **Add to Device Group** step executes.

## Attach Software or Patch Policy Steps

The **Attach Software Policy** and **Attach Patch Policy** steps attach policies to the server, allowing integration with SA Patching and Software Management. These steps can only be executed against a server running a production SA Agent, so if the Build Plan also provisions an OS, these steps must be executed *after* the production OS is running.

## Remediate Server Step

Starts and waits for a remediation job to complete against the server according to the attached policies. Note that the policies will not take effect if this step is not run.

## Managing a Server's State

This section describes managing a server's state which may be required before running Build Plans against the server.

### Asserting the State of the Server

When creating or modifying Build Plans, you must account for the server's state when certain steps are executed. To do this, you use the **Wait for HP SA Agent** script (also called the **Wait** step). The script has parameters that are used to assert the state of the server:

- Using the `--maintenance` parameter asserts that the target server is in Maintenance mode (running a service OS).
- Using the `--production` parameter asserts that the target is in production mode (running a production OS) and has an SA Agent installed with full capabilities.

If the target server is not in the asserted state or if an SA Agent does not report at all, this step fails. This assures that any step after a **Wait** step can make assumptions about the state of the server. For this reason, it is good practice to start Build Plans with a **Wait** step to assert early that the server is in the right state before executing anything else. The next section describes how to force the desired state of the server.

### Changing the State of a Server

You may often need to have a server reboot from the service OS to the production OS or the reverse. To do this, you can use the **Boot** and **Reboot** scripts as **Run Script** steps.

The **Boot** step boots the server into a specific *service OS*. It can make use of all available means to do so. The most generic way to do so requires an SA Agent to be functional on the target server to be able to reboot the server. The **Boot** step configures network booting for this specific server and reboots it to the desired service OS. Note that this only works if network booting occurs *before* booting from disk in the server's boot order.

If the target server is an HP ProLiant with a registered iLO, see [iLO Support](#), the step does not require an SA Agent to be present in order to reboot the server, and also is able to configure one-time boot order on the server, so the server's boot order settings are irrelevant.

## Using Scripts as Building Blocks

The Build Plan steps can be used as building blocks for developing new Build Plans and HP encourages you to use them as building blocks, as you gain knowledge of how Build Plan steps work and interact and become more aware of fixes in newer releases and their impact.

For more information about the included scripts including the `wait`, `boot` and `Reboot` steps, see the descriptions in the SA Client Library.

## Running Build Plans on a Managed Server

While running Build Plans is supported on SA managed servers, most scripts that are installed with SA are designed to work in a Service OS and might not work against a managed server.

The most common reason to run a Build Plan on a managed server is to reboot it into a service OS or perform network personalization.

## Should I Use Server Scripts or OGFS Scripts?

When creating new scripts, you must choose between a *Server Script* or an *OGFS Script*. Both scripts can manipulate files and run commands on the target server. This overlapping functionality can make it difficult to decide which to use.

Typically, server scripts are preferable since distributing work across target servers, rather than doing the work from the SA Core, results in scalable and fast Build Plans since the resources of the target server are used and network round trips are minimized.

Since OGFS scripts run on the SA Core, the load on the SA Core increases as the number of target servers increases and you would need to scale the SA infrastructure in order to use OGFS scripts. For example, if you find that you need more access to the UAPI than is allowed from a target server, you will need to use an OGFS Script.

**Table: Script Types**

Script Type	Run Location	Scalability	UAPI Access
<b>OGFS Script</b>	SA Core	Scales with SA infrastructure	<b>Full:</b> permissions inherited from the user that runs the Build Plan.
<b>Server Script</b>	Local Server	Scales with target servers	<b>Limited:</b> basic operations on the target server



# SA Provisioning Administration

SA Provisioning works right out-of-the-box; however, you may have specific enterprise requirements that you need to configure for SA Provisioning.

## Required Permissions

Before you can configure or perform SA Provisioning, your SA Administrator must have granted you a specific set of SA Provisioning permissions.

You must also have permissions to access the servers associated with SA customers, facilities or server groups. For more information, see the *SA Administration Guide*, Appendix A: “Permissions Reference.”

## SA Provisioning Components

During an SA Core installation, you have the option to install the SA Provisioning components. The following sections describe those components.

### DHCP Configuration (IPv4 and IPv6) for SA Provisioning

The Dynamic Host Configuration Protocol (DHCP) specifies how to assign dynamic IPv4 and IPv6 addresses to servers on a network. SA Provisioning uses DHCP to allow network booting and configuration of unprovisioned and managed servers in the Server Pool. DHCP is also used to configure networking on newly provisioned servers that have not been assigned a static network configuration.

For SA Provisioning, you may use either the DHCP server included with SA or an existing ISC DHCP server. The instructions for configuring these DHCP servers are in the following sections:

- [Configuring the SA DHCP IPv4 Server for SA Provisioning](#)
- [Configuring the SA DHCP IPv6 Server for SA Provisioning](#)
- [Configuring an Existing ISC DHCP Server for SA Provisioning](#)

SA also supports Windows and Linux network booting in DHCPless environments (static IP).

See also [SA Provisioning–Supplied CD Boot Images](#).

### DHCP Software Included with the SA Provisioning Boot Server

When you choose to enable SA Provisioning, the SA Installer installs the provisioning Boot Server and the following:

- **dhcpcd**: An Internet Software Consortium DHCP server (ISC dhcpcd).
- **dhcpcd.conf**: A default DHCP server configuration for IPv4, read by the `dhcpcd` server.
- **dhcpcd\_overrides.conf**: A configuration file for customizations performed by the client for IPv4.
- **dhcpcd\_subnets.conf**: A configuration file with IPv4 subnet details.
- **dhcpcd6.conf**: A default DHCP server configuration for IPv6, read by the `dhcpcd` server.
- **dhcpcd6\_overrides.conf**: A configuration file for customization performed by the client for IPv6.
- **dhcpcd6\_subnets.conf**: A configuration file with IPv6 subnet details.
- **dhcpcdtool; dhcpcd6tool**: The SA DHCP Network Configuration Tool for IPv4 and IPv6, which allows you to modify the `dhcpcd.conf` and `dhcpcd6.conf` files.

### SA DHCP Server (dhcpcd)

The DHCP server will work on the local network only (broadcast domain). To perform SA Provisioning in multiple networks, you must deploy SA Provisioning satellites.

Log messages that the DHCP server produces are sent to the standard UNIX syslog process with the daemon facility. Consult your vendor documentation on how to configure and view syslog messages.

#### SA `dhcpcd.conf` File

The `dhcpcd.conf` file provides the necessary parameters to support network booting of x86 hardware (a PXE-compatible system is required).

The DHCP configuration file is `/etc/opt/opsware/dhcpcd/dhcpcd.conf`. Do not change this file because any changes you make will *not* be migrated during an upgrade. If you must make changes manually in the DHCP configuration files, you should modify the file:

`/etc/opt/opsware/dhcpcd/dhcpcd_overrides.conf` by running the DHCP Network Configuration Tool.

For some advanced configurations (as shown in the section [SA DHCP Network Configuration Tools \(dhcpcdtool and dhcpcd6tool\)](#)), you may need to modify the file with a text editor. Documentation on the DHCP configuration file is available at the ISC web site, <http://www.isc.org>.

The DHCP leases file is `/var/opt/opsware/dhcpcd/dhcpcd.leases`. *This file should not be edited.*

#### SA `dhcpcd6.conf` File

The `dhcpcd6.conf` file provides the necessary parameters to support the IPv6 address assignment for clients.

The DHCP IPv6 configuration file is `/etc/opt/opsware/dhcpcd/dhcpcd6.conf`. In most cases, you will modify this file by running the DHCP Network Configuration Tool for IPv6. For some advanced configurations (as noted in [SA DHCP Network Configuration Tools \(dhcpcdtool and dhcpcd6tool\)](#)), you may need to modify the file with a text editor. Documentation on the DHCP configuration file is available at the ISC web site, <http://www.isc.org>.

The DHCP leases file is `/var/opt/opsware/dhcpd/dhcpd6.leases`. *This file should not be edited.*

### **SA DHCP Network Configuration Tools (dhcpdtool and dhcpd6tool)**

The DHCP Network Configuration Tool is a menu-driven, terminal-based utility that enables you to customize the `dhcpd.conf` and `dhcpd6.conf` files for common network configurations. The tool prompts you for network information needed to configure DHCP for each SA Provisioning network.

Using the DHCP Network Configuration Tool simplifies configuration of the DHCP server and ensures that the DHCP configuration contains the options that are needed for SA Provisioning to function properly.

Additionally, in some environments, multiple IP networks (layer 3) are layered on top of a single VLAN (layer 2). While this configuration is supported by the ISC DHCP server, generally such a topology requires careful consideration to work properly with DHCP. Therefore, the DHCP Network Configuration Tool can only configure a single IP network per VLAN.

The man pages for the DHCP Network Configuration Tool are installed in:

`/opt/opsware/dhcpd/man` on the Boot Server.

### **Required Information for the SA DHCP Network Configuration Tool for IPv4 (dhcpdtool)**

Before you use the DHCP Network Configuration Tool for IPv4 to configure an SA Provisioning network, you need the following information:

- The range of IPv4 addresses that are assigned dynamically by the DHCP server. For example, 192.168.0.11 through 192.168.0.20 might be used to configure a pool of 10 addresses.
- The IPv4 addresses of one or more DNS servers. The DNS servers do not need to be on the same network that is being configured.
- A default DNS domain.

### **Required Information for the SA DHCP Network Configuration Tool for IPv6 (dhcpd6tool)**

Before you use the DHCP Network Configuration Tool for IPv6 to configure an SA Provisioning network, you need the following information:

- The network address and network prefix.
- (Optional): The IPv6 address of one or more DNS servers, which do not have to be on the same network that is being configured.
- (Optional): The range of IPv6 addresses that are assigned dynamically by the DHCP server. If this range is not provided, a range will be selected automatically.

IPv6 SA Provisioning requires that these IPv6 Router Advertisement Flags be enabled:

- M flag - Managed Address Configuration Flag
- O flag - Other Configuration Flag.



dhcpcd6tool checks if the current network has these flags enabled when you add a new network or list current networks.

See RFC 5175 at <http://tools.ietf.org/html/rfc5175> for details about IPv6 Router Advertisement Flags.

## Configuring the SA DHCP IPv4 Server for SA Provisioning

The DHCP Network Configuration Tool for IPv4 is installed with the Boot Server. Perform the following steps to configure networks for SA Provisioning:

1. Log in as root to the server running the Boot Server.
2. (Optional) Make a backup copy of the configuration file with the following commands:

```
cd /etc/opt/opsware/dhcpd
cp dhcpd.conf dhcpd.conf.orig
```

3. Run the DHCP Network Configuration Tool with the following command:

```
/opt/opsware/dhcpd/sbin/dhcpdtool
```

The following DHCP Network Configuration Tool main menu appears:

```
Current Configuration: Full DHCP Management

Select New Configuration:
  1) Full DHCP Management
  2) Disable All DHCP Management
  q) Quit Without Configuration Changes
Choice [1-3, q]: █
```

4. To perform full DHCP Management, enter 1 at the Choice prompt.
5. To add a new network, enter a at the following prompt.

```
Opsware DHCP Network Configuration Tool

a)dd a new network.
e)xit.

Choice [a, e]: █
```

6. To configure the DHCP service on the local network, enter 1 at the following prompt. Local networks are detected automatically and displayed.

```

Opware DHCP Network Configuration Tool

You may add one of the following local network(s) :

1) 192.168.33.0/27 255.255.255.224

Or

r)emote to add remote network.
e)xit to main menu.

Choice [1, r, e]: █

```

- If you are adding a local network, you need to enter the IP addresses or host names of the DHCP range and the DNS servers. In the following two figures, note that the IP addresses are separated by a comma and a space.

```

Opware DHCP Network Configuration Tool

Editing DHCP information for 192.168.33.0/27 (255.255.255.224)

All values which prompt for an address accept either a IP or a hostname.

Enter the DHCP Range (start address, stop address)
: 192.168.33.3, 192.168.33.23
Enter the DNS server(s) (comma separated)
: 192.168.162.139, 192.168.163.142
Enter the DNS domain: opsware.com
Would you like to add the IPs from DHCP range in /etc/hosts ? (y/n): █

```

```

Opware DHCP Network Configuration Tool

Editing DHCP information for 192.168.33.0/27 (255.255.255.224)

1) gateway      : 192.168.33.1
2) DHCP range  : 192.168.33.3 - 192.168.33.23
3) DNS servers: 192.168.162.139, 192.168.163.142
4) DNS domain  : opsware.com
5) Power6 provisioning override: No
   !!! WARNING: Option 5) breaks Solaris SPARC OS provisioning !!!
   !!! Make sure to deactivate this option before trying to provision a Solaris SPARC machine !!!

1..5 to edit option.
d)delete network and return to main menu.
k)keep network and return to main menu.

Choice [1..5, d, k]: █

```

- If the displayed information is correct, enter **k** to keep the network and return to the main menu.
- At the main menu, to save the information you have entered, enter **s**  
or  
to edit a configured network, enter the corresponding integer and return to step 3

or

to add more networks, enter `a` and return to step 3.

10. To exit the DHCP Network Configuration Tool, enter `e`. You are prompted to start (or restart) the DHCP server process.
11. To start (or restart) the DHCP server process, enter `y`. The DHCP Network Configuration Tool displays diagnostic output as part of its startup.

## Configuring the SA DHCP IPv6 Server for SA Provisioning

The DHCP Network Configuration Tool for IPv6 is installed with the Boot Server. Perform the following steps to configure networks for OS provisioning:

Log in as root to the server running the Boot Server.

(Optional) Make a backup copy of the configuration file using the following commands:

```
cd /etc/opt/opsware/dhcpd
cp dhcpd6_subnets.conf dhcpd6_subnets.conf.orig
```

Run the DHCP Network Configuration Tool with the following command:

```
/opt/opsware/dhcpd/sbin/dhcpd6tool --help
```

```
[root@mars /]# /opt/opsware/dhcpd/sbin/dhcpd6tool --help
usage: dhcpd6tool [-h] [-r] [--dry-run] [--no-ra-check]
                {add,del,edit,clear,list} ...

Run with dhcpd6tool <subcommand> --help for more information about each
subcommand

optional arguments:
  -h, --help            show this help message and exit
  -r, --restart          restart the daemon after applying changes
  --dry-run             do not read or write anything. Useful for previewing
                        automatic values.
  --no-ra-check         do not check for Router Advertisement messages on the
                        network. Only disable if you know what you are doing.

subcommands:
  {add,del,edit,clear,list}
    add                 add new subnet
    del                 remove existing subnet
    edit                overwrite an existing subnet configuration
    clear               clear all configured subnets
    list                list configured subnets
[root@mars /]# █
```

To add a new network or modify an existing network, it is mandatory that you provide a network address and prefix. Optional parameters: DNS list and IPv6 address range.

If the *DNS server list* is not provided, it is detected automatically by the configuration file:  
`/etc/resolv.conf`.

If an *IPv6 address range* is not provided, it is detected automatically based on the current *network address* and *prefix* parameters, and a chunk from that address space will be used.

For example:

```
/opt/opsware/dhcpd/sbin/dhcpd6tool --restart add
fc00:508:1:0::0/64 -i fc00:508:1:0:666::/120 -n fc00:302:1::1
```

This example shows:

Network address and prefix: `fc00:508:1:0::0/64`

IPv6 address range: `fc00:508:1:0:666::/120`

DNS server address: `fc00:302:1::1`

1. Log in as root to the server running the Boot Server.
2. (Optional) Make a backup copy of the configuration file using the following commands:

```
cd /etc/opt/opsware/dhcpd
cp dhcpd6_subnets.conf dhcpd6_subnets.conf.orig
```

3. Run the DHCP Network Configuration Tool with the following command:

```
/opt/opsware/dhcpd/sbin/dhcpd6tool --help
```

```
[root@mars /]# /opt/opsware/dhcpd/sbin/dhcpd6tool --help
usage: dhcpd6tool [-h] [-r] [--dry-run] [--no-ra-check]
                {add,del,edit,clear,list} ...

Run with dhcpd6tool <subcommand> --help for more information about each
subcommand

optional arguments:
  -h, --help            show this help message and exit
  -r, --restart          restart the daemon after applying changes
  --dry-run             do not read or write anything. Useful for previewing
                        automatic values.
  --no-ra-check        do not check for Router Advertisement messages on the
                        network. Only disable if you know what you are doing.

subcommands:
  {add,del,edit,clear,list}
  add                  add new subnet
  del                  remove existing subnet
  edit                 overwrite an existing subnet configuration
  clear                clear all configured subnets
  list                 list configured subnets
[root@mars /]# █
```

4. To add a new network or modify an existing network, it is mandatory that you provide a

network address and prefix. Optional parameters: DNS list and IPv6 address range.

- If the *DNS server list* is not provided, it is detected automatically by the configuration file: `/etc/resolv.conf`.
- If an *IPv6 address range* is not provided, it is detected automatically based on the current *network address* and *prefix* parameters, and a chunk from that address space will be used.

For example:

```
/opt/opsware/dhcpd/sbin/dhcpd6tool --restart add
fc00:508:1:0::0/64 -i fc00:508:1:0:666::/120 -n
fc00:302:1::1
```

This example shows:

- Network address and prefix: `fc00:508:1:0::0/64`
- IPv6 address range: `fc00:508:1:0:666::/120`
- DNS server address: `fc00:302:1::1`

## Starting and Stopping the SA DHCP Server for IPv4 and IPv6

To start the DHCP server process, enter the following command on the server running the Boot Server:

```
/etc/init.d/opsware-SA start dhcpd
```

```
/etc/init.d/opsware-SA start dhcpd6
```

To stop the DHCP server process, enter the following command on the server running the Boot Server:

```
/etc/init.d/opsware-SA stop dhcpd
```

```
/etc/init.d/opsware-SA stop dhcpd6
```

### Modifying the `dhcp.conf` File for Use with WINPE

Typically, networks with other DHCP servers do not have the SA DHCP server configured as authoritative. However, WINPE requires that the DHCP server be authoritative in order for servers to be able to boot using WINPE.

The `dhcp.conf` file provided with SA 9.0 by default has the `authoritative` setting commented out. If you need to boot servers using WINPE, you will need to uncomment this line:

1. Log on to the Core's dhcpd server as root.
2. cd to the `/etc/opt/opsware/dhcpd` directory.
3. Issue the following command:

```
chmod a+w dhcpd.conf
```

4. Open an editor and edit the `dhcp.conf` file; for example:

```
vi dhcpd.conf
```

and add or uncomment `authoritative`.

5. Save the file.
6. Issue the command:

```
chmod a-w dhcpd.conf
```

7. Issue the command:

```
/etc/init.d/opsware-SA restart dhcpd
```

### Configuring an Existing ISC DHCP Server for SA Provisioning

To configure an existing ISC DHCP server, perform the following steps:

1. Ensure that the configuration file for the existing ISC DHCP server is a copy of configuration file that SA provides, which is located in `/etc/opt/opsware/dhcpd/dhcpd.conf`.
2. The SA DHCP server must *not* be running on the server hosting the Boot Server.
  - To disable SA DHCP for **IPv4**, use the `dhcpdtool` and select the **Disable All DHCP Management** option (this preserves the configuration).
  - To disable SA DHCP for **IPv6**, run the commands:
 

```
/opt/opsware/dhcpd/sbin/dhcpd6tool --no-ra-check disable  
/etc/init.d/opsware-SA stop dhcpd6
```
  - To delete SA DHCP **IPv6** configuration, run the command:
 

```
/opt/opsware/dhcpd/sbin/dhcpd6tool --restart clear
```
3. Ensure that the DHCP configuration for systems to be provisioned has all required details, such as the DNS server, netmask, default router, DNS domain, and so forth.
4. Restart the existing ISC DHCP server.

### Configuring a Windows DHCP Server for SA Provisioning

You can use a Microsoft Windows DHCP server instead of the Opware-supplied DHCP server to provision both Windows or Linux on PXE 2.0 clients.

The Microsoft Windows DHCP server cannot be used during the SA Provisioning of the following types of systems:

- PXE 0.99 or 1.x clients (These older PXE clients have old PROMS and a PXE bootstrap floppy made with `rbfg.exe`.)
- Solaris

**Note:** The Windows DHCP server only responds with options explicitly requested by the client. As options 252 is not part of the standard PXE boot request, the Windows DHCP server does not supply it.

When provisioning a 64 bit Windows OS, in order for the x64 operating system image to be selected by default, the SA DHCP server sends a specific string to the client in option 252. The value of this option is set to the string: BootBCD.

If this option is not sent back to the client in the DHCP OFFER response the WinPE OS selection will default to the x86 image.

To configure a Microsoft Windows DHCP server for use with SA Provisioning, perform the following tasks:

1. On the Windows system running the DHCP server, you must define option #60, so that it appears in the DHCP scope options. To do so, open a command prompt window, and enter the following command:
 

```
netsh.exe dhcp server add optiondef 60 "PXEClient" STRING
```
2. Using the Windows DHCP Management Snap-in (`dhcpcmgmt.msc`), create a scope, which is usually a subnet declaration. In the scope options, #60 should now appear. Check the box, and then add the string `PXEClient`.
3. Using the same scope options box, configure options 66 and 67: Click the DHCP option #66 (Boot Server Host Name), and add the full DNS name of the TFTP/Boot Server (for example, `core01.test.com`). For option #67 (Bootfile Name), add the boot file name: `pxelinux.0`.
4. Ensure that the DHCP scope for the systems to be provisioned is configured with the required details, such as the DNS server, netmask, default router, DNS domain, and so on.
5. At the command prompt, enter the following commands to define the IP address of the Agent Gateway and the port forward for the Build Manager:
 

```
netsh.exe dhcp server add optiondef 186 "buildmgr_ip"
IPADDRESS

netsh.exe dhcp server add optiondef 187 "buildmgr_port" WORD
```
6. Using the DHCP Management Snap-in (`dhcpcmgmt.msc`), configure options 186 and 187 to be part of your scope, and give them the appropriate values (IP address of the Agent Gateway and the port forward for the Build Manager, normally 8017).
7. Define option 043 (Vendor specific options) as a BINARY type, with the value `01 04 00 00 00 00 ff`. This setting tells the DHCP server to go directly to the FTP server specified in the Boot Server Host Name parameter, and also tells it to not use Multicast TFTP.
8. Restart the Windows DHCP server.

## Controlling the SA and Windows DHCP Servers' Responses to SA Provisioning Requests

You can configure the SA DHCP server to respond only to the SA Provisioning requests from PXE and Sun Solaris JumpStart clients while the Microsoft Windows DHCP server responds to all Windows provisioning requests.

1. Add the network subnet to the SA DHCP server (see [Configuring the SA DHCP IPv4 Server for SA Provisioning](#)).

2. Stop the SA DHCP server:

```
/etc/init.d/opsware-SA stop dhcpd
```

3. Make a backup copy of the SA DHCP configuration file:

```
cd /etc/opt/opsware/dhcpd
cp dhcpd.conf dhcpd.conf.orig
```

4. In a text editor, open the SA DHCP configuration file.

5. Below the `pool` entry, find the subnet definition you want to configure and comment it out with the `#` character:

```
range <IP1> <IP2>;
```

Should now read:

```
# range <IP1> <IP2>;
```

6. Immediately after the now commented out range line, enter:

```
pool {
# range <IP1> <IP2>;
allow members of "solaris-sun4u";
allow members of "solaris-sun4us";
allow members of "pxeclients";
range <IP1> <IP2>;
}
```

modifying the above as necessary to fit your system. The `pool` statement tells the DHCP server to continue serving the specified range, but only for the three types of clients indicated. (The first two `allow` statements are for Sun machines, the third is for PXE clients). The closing brace in the `pool` statement is required.

7. Repeat the preceding two steps for every subnet you wish to configure.

8. In the text editor, save the `dhcpd.conf` file.

9. Start the SA DHCP server:



```
/etc/init.d/opsware-SA start dhcpd
```

10. Check the DHCP logs for errors. The DHCP service logs with `syslog`. See the `syslog.conf` file to determine how logging has been configured for the SA DHCP server.
11. Ensure that the Windows DHCP server subnet/scope declarations are modified to include the Build Manager DHCP options (code 186 and 187). See [Configuring a Windows DHCP Server for SA Provisioning](#).
12. Ensure that the Windows DHCP server does not include options 43, 60, 66, or 67 in the scope/subnets. This will prevent the PXE and Sun JumpStart clients from connecting to the Windows DHCP server but allow them to connect to the SA DHCP server.
13. Ensure that the IP ranges of the Windows and SA DHCP servers do not overlap. As a guideline, the number of IP addresses in a given range should be twice the maximum number of servers that will be provisioned concurrently.
14. If the DHCP servers are not directly connected to the network/subnet of the systems being provisioned, the DHCP requests must be forwarded to both DHCP servers, the SA DHCP server first.

### Enabling IBM POWER6 SA Provisioning with the DHCPD Tool

Due to hardware constraints, IBM POWER6 servers cannot be provisioned using the normal SA Provisioning procedures. However, you can enable provisioning for these servers using the DHCP Tool provided with SA.

This involves using `dhcpcdtool` to set a parameter in the initial subnet declaration, which enables Linux SA Provisioning on the IBM POWER6 server's hardware. This workaround also requires that SA be in Full DHCP Management mode.

**Note:** Setting this parameter prevents Solaris SPARC SA Provisioning. You will see a warning message to this effect when you change the parameter.

Perform the following tasks to enable Linux SA Provisioning on IBM POWER6 hardware:

1. Start `dhcpcdtool`.
2. Select "Full DHCP Management."
3. Select the network for which the workaround is to be applied. An editing menu similar to the following displays:

```
Opsware DHCP Network Configuration Tool
```

```
Editing DHCP information for 192.168.208.160/27
(255.255.255.224)
```

```
1) gateway : 192.168.208.161
```

```
2) DHCP range : 192.168.208.163 - 192.168.208.190
```

```
3) DNS servers: 192.168.194.4
4) DNS domain : dev.opsware.com
5) Power6 provisioning override: No
!!! WARNING: Option 5) breaks Solaris SPARC OS provisioning
!!!
!!! Make sure to deactivate this option before trying to provision a
Solaris SPARC machine !!!
1..5 to edit option.
d)delete network and return to main menu.
k)keep network and return to main menu.
Choice [1..5, d, k]:
```

4. Select option 5 and press ENTER.
5. The message "Are you sure that you want to toggle POWER6 provisioning although it breaks Solaris SPARC provisioning (Enable/Disable)?" displays.
6. Press E to Enable POWER6 provisioning or D to Disable, then select K to keep the settings then S to save them.
7. Exit the DHCPD tool. Upon exiting, the DHCPD server is restarted and the new setting takes effect.

**Note:** If you are not using the SA-provided DHCPD server, you must add the following line to the subnet in which POWER6 will perform SA Provisioning:

```
filename "yaboot";
```



# OS Sequence-Based Provisioning Requirements, Setup, and Usage

**Deprecation Notice:** For certain managed server platforms, OS Sequences are deprecated in SA 10.10 and later and have been replaced by the new and more powerful provisioning Build Plans. See the *SA Support and Compatibility Matrix* for the platforms that support Build Plans. It is strongly recommended that you migrate any existing OS Sequences to Build Plans on these platforms. Any new SA provisioning templates you need to create on these platforms should use Build Plans.

## The OS Sequence Provisioning Process

Using SA OS Sequences requires 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.
  - Uploading licensed OS media to the Media Server
  - DHCP server management using the:  

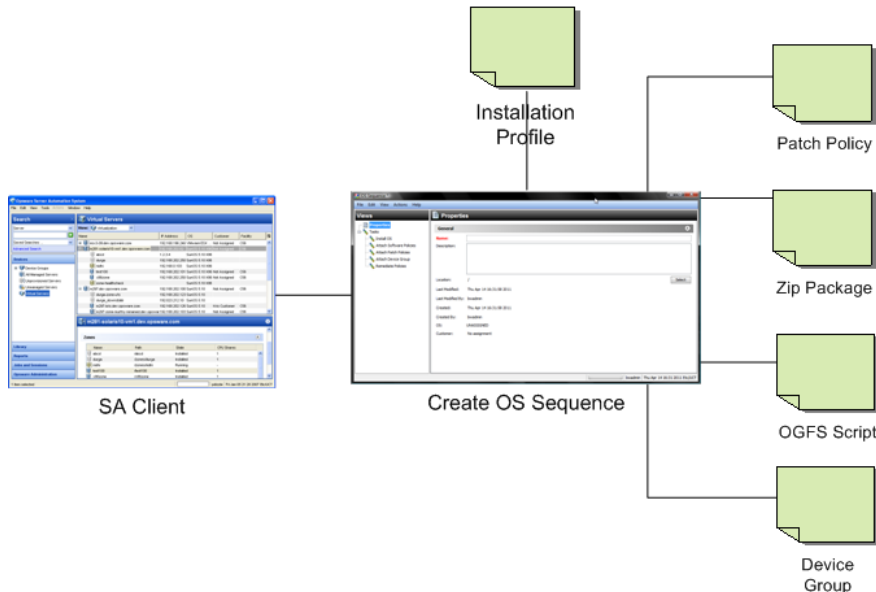
```
/opt/opsware/dhcpd/sbin/dhcpdtool.
```
- Creating OS Installation Profiles for the operating systems to be provisioned. The Installation Profiles specify which operating system is to be installed and how it is to be configured and where the operating system media is located on the Media Server (using an MRL).

Associated with the Installation Profile are:

- Operating system-specific installation configuration files such as Kickstart (Linux), Jumpstart (Solaris/SPARC 10), Automated Installer (Solaris/SPARC 11) 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.

- 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.

## Create OS Sequence



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

### 1. Preparation

1. Physically prepare the server for operation and connect it to a network that can communicate with SA.
2. In some cases, you must prepare the server hardware for OS Provisioning.
3. OS Installation Profile(s) 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:

1. Use a bootable image provided by SA.

**Note:** A bootable CD or DVD is not required for Intel-based servers that support PXE/WinPE or UNIX servers as these servers can be remotely booted over a network.

2. 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 [Network Booting](#) and [Managed Boot Clients](#).

3. A bootable CD or DVD is not required for Intel-based servers that support PXE/WinPE or UNIX servers as these servers can be remotely booted over a network.
4. After the server boots successfully and the SA Build Agent has been installed, the server appears in the SA Client in the Unprovisioned Server list and is ready for operating system installation.
5. **Install the Operating System (Provision)**

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.
6. Start the OS Provisioning job.

## SA OS Provisioning Components

SA 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 Installation Guide*.

SA OS Sequence Provisioning uses the following components and features.

- The OS Build Agent
- The Build Manager
- The Media Server
- The Boot Server

### The OS Build Agent

Used with OS Sequences, the Build Agent is Similar to the SA 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 with an SA-supplied image 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.

### The Build Manager

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 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

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
```

### Oracle Solaris/Sun SPARC 10 and 11

For Oracle Solaris/Sun SPARC 10 and 11 OS Provisioning, the JumpStart build script (Solaris/SPARC 10) or Automated Installer (Solaris/SPARC 11) 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](#) and [Booting a Windows Server in a Non-DHCP Environment](#).

### Provisioning Setup for OS Sequences

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 specified for users who will perform OS Sequence management and execution.
- Confirming that the network is configured as required for SA 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.
- Creating Media Resource Locators (MRLs) that identify the location of the media during provisioning.
- Configuring optional HP RAID configuration capture.
- 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 Sequences that specify the order of provisioning tasks and can include optional configuration information such as Software Policies, Windows Patch policies, Static Device Groups.
- 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 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 Installation Guide*.
4. Ensuring that you have the correct permissions for OS Sequence management and 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:
  1. 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.

2. 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. 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 value.

For OS Sequence-based provisioning, custom attributes can be added to OS Installation Profiles.

9. Creating OS Sequences, which allow you to specify provisioning tasks, the Installation Profile to use, reboots, etc. While this task can be done by the Provisioning Administrator, it can also be delegated to users who perform provisioning.
10. 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

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 SA Provisioning. You must prepare the Media Server by uploading your images to the Media Server.

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 \(MRLs\)](#). 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.

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>@@MediaServer@/OS Media
```

The username/password must be preceded by “@@” and ended by “@”. For example:

```
smb://user4312.example.com\thsu_usr:-smith123!@@MediaServer@/OS Media
```

- 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: Table 3. Import Media Tool Command Line Options**

<b>Import Media Tool Option</b>	<b>Description</b>
<code>--help</code>	Display this help.
<code>--folder</code>	Override Folder location. Default is: “/Package Repository/OS Media/<Platform Name>”.
<code>--media-name=&lt;displayname&gt;</code>	Override automatically-generated display name. Note: Use ‘_’ to escape spaces in the name.
<code>--removeMRL=&lt;mrlid&gt;</code>	Remove existing Media Resource Locator from HP SA specified by mrlid
<code>--listMRLs</code>	List Media Resource Locators available in HP SA
<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&lt;mrl&gt;</code>	Override automatic OS Media path generation. <code>--mrl- =//MEDIA/PUB/WINNT/SERVER/I38- 6  --mrl- =nfs:/- /media/export/media/redhat/7.2</code>
<code>--smbuser=&lt;user&gt;</code>	User for SMB access. Default is “root”.
<code>--smb-password=&lt;password&gt;</code>	Use this password for SMB access. <b>Note:</b> This appears in cleartext on the command line. <b>Warning:</b> 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

Import Media Tool Option	Description
	securely.
<code>--logfile=&lt;logfile&gt;</code>	Override log file location. Default is: <code>/var/log/opsware/mm_wordbot/import_media.log</code>
<code>--wimimage</code>	The path supplied refers to a (WIM) image. Be sure to also supply <code>--platform=&lt;platform&gt;</code> , since the target platform cannot be autodetected.
<code>--platform=&lt;platform&gt;</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, 2008 R2 x64, and 2012 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 Server 2003, 2008 and 2008 R2 x64, 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 in 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 2K Server Using a Share**

You want to host your Windows media on a Windows 2K (2003, 2008, 2008 R2 x64) 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/opsware/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 user-name/password:

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

**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/opsware/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

1. 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
```

2. Copy the contents of the first SUSE Linux 9 CD1 to the sles9/suse/CD1 directory.

**Note:** 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.

3. Copy the contents of the second SUSE Linux 9 CD2 to the sles9/core/CD1 directory.

4. 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.

5. 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
```

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

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

7. 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:

1. Add the following directories:

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

2. Copy the contents from the SP3 CD1, CD2, and CD3 to `sles9/CD1`, `sles9/CD2`, and `sles9/CD3`, respectively.
3. 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
```

4. 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](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](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](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](http://www.suse.com/~ug/autoyast_doc/index.html)

### **Configuring the Media Server for Oracle Sun Solaris 10**

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/opsware/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.****Configuring the Media Server for Oracle Sun Solaris 11****1. On the Media Server, create the directory**

```
/media/opsware/sunos/Solaris11
```

**2. Download the Solaris 11 images to the directory above.**

There are two ISO images:

- Part A, x86 (3.3 GB)
- Part B, x86 (3.1 GB)

**3. Follow the instructions on the download page to concatenate the two images into a single full image.****4. Create a symlink (symbolic link) to the repository on the Media Server. This symlink is required when provisioning Solaris 11 x86 and Solaris 11 SPARC on the same core because both operating systems are contained in the same image and a Media Resource Locator (MRL) cannot be created using the same path for two operating systems.**

For example:

```
# cd /media/opsware/sunos/
# ls -lsa
8 drwxr-xr-x 3 root root 4096 Mar 26 14:33 solaris11_repo
#ln -s solaris11_repo solaris11_link
# ls -lsa
4 lrwxrwxrwx 1 root root 15 Mar 26 14:39 solaris11_link ->
solaris11_repo/
8 drwxr-xr-x 3 root root 4096 Mar 26 14:33 solaris11_repo
```

**Oracle Solaris Automated Installer**

The Oracle Solaris Automated Installer uses two response files:

- `ai.xml`: for the Automated Installer to specify partitioning, locales, source repository and packages to be installed

- `sc.xml`: for System Configuration to specify host name, user passwords and specific configurations, services configurations, and network configurations.

**Note:** Oracle documentation about creating `ai.xml` and `sc.xml` files can be found at:

[http://docs.oracle.com/cd/E23824\\_01/html/E21798/](http://docs.oracle.com/cd/E23824_01/html/E21798/)

Since SA can handle only one response file per operating system and the Solaris Automated Installer requires two files for provisioning Solaris 11, SA provides a script, `join_ai_sc.py`, located in `/opt/opsware/buildscripts/solaris/tools`, on a Slice component bundle server, that can join `ai.xml` and `sc.xml` into a single file. The output file this script creates is then used as the response file for Oracle Solaris 11 provisioning.

### Choosing Between Solaris 10 SPARC and Solaris 11 SPARC

Because Oracle Sun SPARC provisioning uses the `bootp` protocol, there is no clear way to choose between Solaris 10 SPARC and Solaris 11 SPARC provisioning. Therefore, you must use the `/opt/opsware/boot/jumpstart/tools/switch_solaris_sparc_miniroot` command to tell SA which version of Solaris to choose. This tool can be found on the core server on which the OS Provisioning component bundle is installed.

### Enabling Oracle Solaris 11 x86 with the Manage Boot Client

By default, the Manage Boot Client (MBC) utility can only automatically provision Solaris 10 X86. To enable Oracle Solaris 11 X86 in MBC, run the tool `/opt/opsware/boot/js-x86/tools/switch_solaris_x86_default_pxe` and choose the desired default option from the Solaris Preboot Execution Environment (PXE) menu. Once Solaris 11 is set as the default, all machines that boot into the Server Pool will boot the Solaris 11 X86 miniroot. Also all MBC jobs will use the Solaris 11 X86 miniroot, causing MBC jobs for Solaris 10 X86 to fail. This tool is found on the core server on which the OS Provisioning component bundle is installed.

### 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 an 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
```

To import Solaris 11 SPARC media from a NFS server:

```
import_media nfs://mediaserver.company.com/export/media/solaris/solaris11_repo/repo
```

To import Solaris 11 x86 media from a NFS server:

The Solaris 11 repository is detected by default as Solaris 11 SPARC media. Therefore, to use this repository to provision a Solaris 11 X86 server, you must run import\_media with --platform="SunOS 5.11 X86" parameter.

```
import_media --platform="SunOS 5.11 X86" nfs://-
mediaserver.company.com/export/media/solaris/solaris11_link/repo
```

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](#).

## Media Resource Locator Administration

### Deleting MRLs

You cannot delete an MRL 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.

To delete an MRL use the `--removeMRL` argument of the `import_media` script followed by the ID of the desired MRL to be removed. To find out the ID of the MRL use the `--listMRLs` argument of the `import_media` script.

### 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.

## Multipath SAN Support for OS Provisioning

SA provides multipath SAN support for the topics listed in this section.

## OS Sequences

Red Hat 6 automatically identifies multipathing and enables kernel modules, but Red Hat 5 does not.

For Red Hat 5, before you run a Red Hat Enterprise Linux (RHEL) 5 OS Sequence, use the following steps to pass `MPATH` as a kernel argument for the server:

1. Define a custom attribute `kernel_arguments` for the server, and set its value to `mpath`. See [Custom Attributes for Linux or VMware ESX](#).
2. Create the OS Installation Profile.
3. Add a custom attribute to the Installation profile, and set its value to `mpath`.
4. Set the value of the installation profile to `mpath`.

If you are performing multipath installation, it is also recommended that you add `mpath` as the kernel argument in the Red Hat 5 kickstart file. You can do this by creating a copy of the Configure Default Red Hat 5 `ks.cfg` and modifying the file.

### SUSE Linux Enterprise Server 11

SUSE Linux Enterprise Server 11 does not identify multipathing by default. Use the procedures in this section to install SUSE Linux Enterprise Server on multipath LUNs.

### Updating Multipath Drivers

You must copy the drivers into the media to enable multipathing. More information about the driver update disk (DUD) and the process of updating the media is available here:

<http://www.novell.com/support/viewContent.do?externalId=7009981&sliceId=1>

To configure SA, you must perform the following tasks:

1. Download the driver update disk (DUD) from the above link.
2. Extract the contents of the DUD to the SUSE Linux Enterprise Server 11 media.
3. Upload the SUSE Linux Enterprise Server 11 media to the SA Media Server using the `import_media` command.
4. If you already have the media uploaded into the SA Media Server, extract the contents of the DUD as directed by the SUSE support document.

**Note:** After you complete these steps, multipathing will be enabled for all subsequent installations. To perform non-multipathed installations, create a separate SUSE media within SA Media Server.

### Defining Kernel Arguments

To enable multipath installation, pass `mpath` as a value for the custom attribute `kernel_argument` by doing one of the following:

- When creating the OS Installation Profile, define the custom attribute `kernel_argument` and set its value to `mpath`.

- Before running the SUSE Linux Enterprise Server 11 OS sequence, define a custom attribute `kernel_argument` on the server to be provisioned, and set its value to `mpath`.

### Partitioning Section in AutoYaST

SUSE Linux Enterprise Server 11 requires that devices be specified in Mapper format during multipath environment installations. SA OS Provisioning verifies the AutoYaST profile and modifies it to suit the multipathing environments, based on the following rules:

**Table:**

Case	SA Solution
1. AutoYaST does not have the partitioning section.	SA adds the partitioning section with the drive and device.
2. AutoYaST has a single drive section, but does not have a device section.	SA inserts the boot drive expressed in Mapper format.
3. AutoYaST has multiple drive sections, but does not have device sections. Has multiple drive sections but no <code>&lt;device&gt; in &lt;drive&gt;</code> section.	SA does not make any changes to AutoYaST. It prints a warning and continues with the installations.  This type of configuration can cause installation failure. To ensure a successful installation, complete each of the drive sections with devices in Mapper formats.
4. AutoYaST has devices in the configuration <code>Single_OR_multiple drive sections and a &lt;device&gt; specified for the &lt;drive&gt;</code> .	SA checks if the drive attributes in AutoYaST are in the correct Mapper format. If they are not, SA prints a warning and makes no further checks for correctness.  If the drive attributes are not in the correct format, the installation fails. To ensure a successful installation, complete each of the drive sections with devices in Mapper formats.

### Friendly Device Name

`/dev/mapper` devices should only be used during installation. The SUSE Linux Enterprise Server Storage Administration Guide states that, in the running system, the multipath devices should be accessed through `/dev/disk/by-id/`.

SA sets up the scripts to convert the devices from `/dev/mapper` to `/dev/disk/by-id` when you define the custom attribute `friendly_mpath_device` to `true`.



## Windows 2008/ Windows 2008 R2

Windows supports installations in multipath environments. Install multipath Device Specific Modules (DSMs) to further improve the capabilities of the target server. Import these DSMs into the SA Software Repository and install them using software policies attached to OS sequences.

You can modify the SAN Policy for the Windows Service OS (Winpe32, winpe64, winpe32-ogfs, winpe64-ogfs) shipped with SA. For more information about available options, see:

[http://technet.microsoft.com/en-us/library/cc749466\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc749466(v=ws.10).aspx)

The default SAN Policy registry value in the SA-provided Windows pre-installation environment (WinPE) boot image is 1.

## VMWare ESX/ESXi

VMWare ESX and ESXi support multipathing - no additional steps are required.

## 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

**Note:** 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.

**Note:** 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.

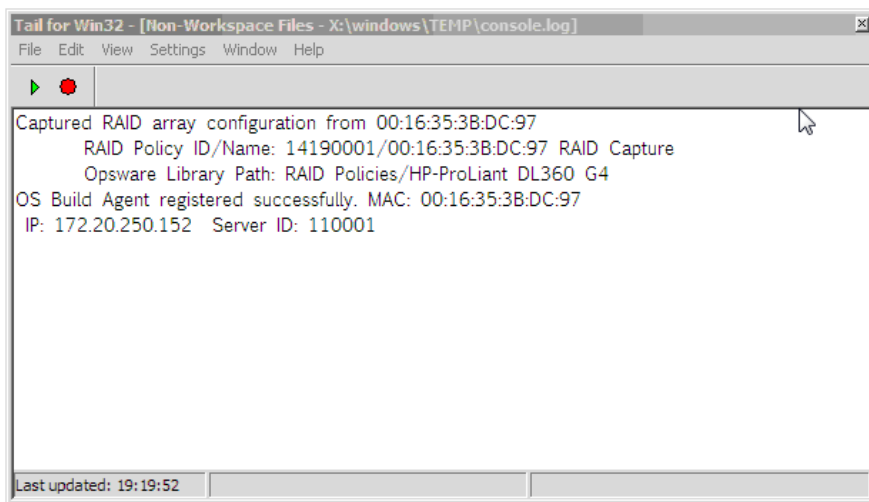
You can do this in either of two ways:

1. 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 Option](#) for information on creating or modifying a server record with MBC.
2. 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 6** in the console.log file:

### Figure 6. RAID Configuration Capture Message

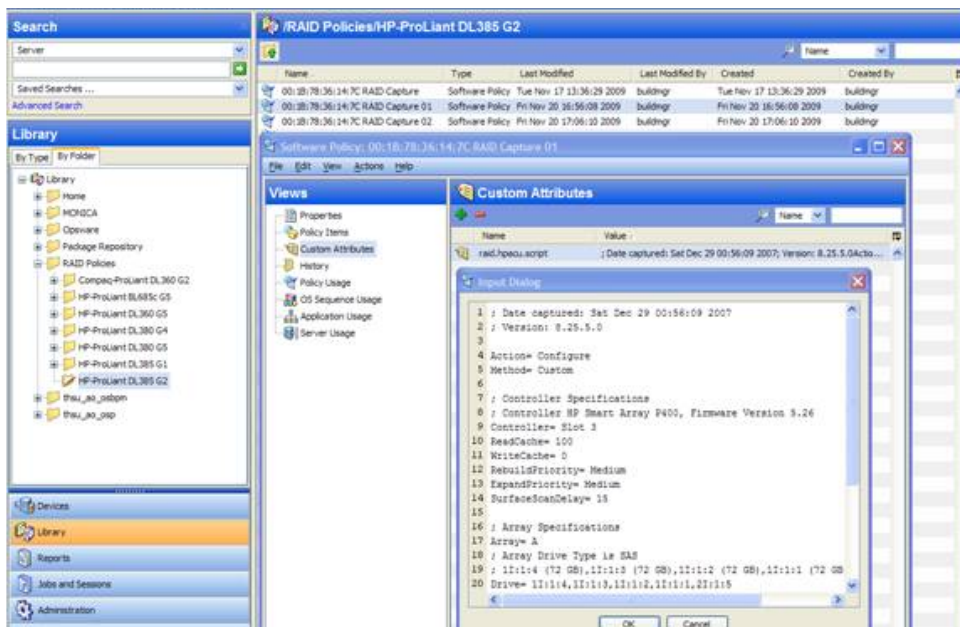


By default, the software policy is given a name that consists of the server's MAC address appended with the words `RAID Capture`, such as `00:16:35:3B:DC:97 RAID Capture`. You can rename the file in the SA Client. After the RAID configuration is captured, the value of the custom attribute `raid.capture` 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 7. 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.

**Note:** 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.

**Note:** If SA fails to configure an HP RAID controller during a Run OS Sequence job, a subsequent attempt to capture the HP RAID controller configuration may fail with the following message:

```

RAID configuration deployment failed: Failed to deploy RAID
configuration: An error occurred while clearing current array
configuration. Exit status: 1280
Error message from ACU: ERROR: (2821) No controllers detected.
  
```

This is due to a known issue in the HP ACU controller. In this case, you must manually configure the HP RAID controller with a logical volume at server boot time.

## 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.

**Note:** 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.

## Defining Installation Profiles and OS Sequences

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

- Installation Profiles
- [Creating OS Sequences](#)

This chapter describes how to define and administer these files.

## OS Installation Profile Requirements

This section discusses the following topics:

- [Overview](#)
- [Specifying Software for OS Provisioning](#)
- [Configuration Files](#)
- [Oracle Solaris/Sun SPARC 10 Installation Profile Requirements](#)
- [Red Hat Linux Installation Profile Requirements](#)
- [VMware ESX Installation Profile Requirements](#)
- [SUSE Linux Installation Profile Requirements](#)
- [Microsoft Windows Installation Profile Requirements](#)

### Overview

You use OS Installation Profiles in conjunction with OS Sequences. 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 Installation Guide* and in [Creating Media Resource Locators \(MRLs\)](#) and have created MRLs pointing to the operating system media using the Import Media tool as described in [Setting Up the Media Server](#).

You create OS Installation Profiles by using the `install_profile` script which is installed with SA's Software Repository component.

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 where to perform an installation by selecting an MRL. (See [Media Resource Locator Administration](#) 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](#) for more information.)
  - A build customization script, which can modify the installation process at certain points. (See [Creating Build Customization Scripts](#) for more information.)
  - Microsoft Windows Only:
    - A Hardware Signature, which contains hardware specific information.
    - A partitioning disk script file, containing partitioning procedures
    - A formatting disk script file, containing formatting procedures

**Table 5** compares the installation resources across operating systems.

Table 5: Installation Resources for OS Installation Profiles

<b>Installation Resource</b>	<b>SUSE</b>	<b>Windows Server 2003</b>	<b>Windows Server 2008/2012</b>	<b>Solaris/SP ARC 10</b>	<b>Solaris/SP ARC 11</b>	<b>Linux or VMware ESX</b>
Con-figuration File	YAST profile autoinst.xml	unattend.txt	unattend.xml	Jumpstart profile	Automated Installer	Kick-start/Weasel profile
Build Customization Script	Optional executable file: bcs.tgz	Optional executable file: WinPE:	Optional executable file: WinPE:	Optional executable file: bcs.tar.Z	Optional executable file: bcs.tar.Z	Optional Executable file: bcs.tgz containing

Installation Resource	SUSE	Windows Server 2003	Windows Server 2008/2012	Solaris/SPARC 10	Solaris/SPARC 11	Linux or VMware ESX
	containing "run" script	bcs.zip containing run-phase.bat script	bcs.zip containing run-phase.bat script	containing run script	containing run script	"run" script
Hardware Signature File	Not required	Optional file-name.txt	Optional file-name.txt	Not required	Not required	Not required

**Note:** 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](#).

## 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:

- Oracle Solaris/Sun SPARC 10  
JumpStart profile
- Oracle Solaris/Sun SPARC 11  
Automated Installation
- 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`

**Note:** 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 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`.

## Oracle Solaris/Sun SPARC 10 Installation Profile Requirements

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

The Solaris/SPARC 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.

See [Sample Response File for Windows Server 2003](#) below for an example of a valid Windows response file.

#### **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 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

```

## Defining and Managing OS Installation Profiles

The `install_profile` script is used to define OS Installation Profile. You must log in as an SA user (username and password) that has the required permissions to create or modify OS Installation Profiles.

This section includes the following topics:

- [Install Profile Tool Syntax and Options](#)
- [Defining an OS Installation Profile — Linux/UNIX](#)
- [Defining an OS Installation Profile — Windows](#)
- [Modifying Existing OS Installation Profiles](#)
- [Changing the OS Installation Profile Properties](#)
- [Modifying the OS Installation Profile Packages](#)
- [Viewing Change History for an OS Installation Profile](#)
- [Deleting an OS Installation Profile](#)

### Install Profile Tool Syntax and Options

This section provides the syntax and command line options for the Install Profile Tool. To start the tool, log in to the Software Repository server (Slice Component bundle host) and enter **install\_profile [options]**.

Install Profile Tool Command Line Options

Command line options	Description
<code>--help</code>	Displays help
<code>create</code>	Create an OS Install Profile

<b>Command line options</b>	<b>Description</b>
update	Update an existing OS Install Profile
remove	Remove an existing OS Install Profile
listMRLs	List the current Media Resource Locators available in SA
listPlatforms	List platforms available in SA. Note that not all platforms are available for OS Sequence Provisioning
--user	SA user name that will operate on the OS Install Profile (you must have permissions to interact with OS Install Profiles)

Install Profile Tool **create** specific options

<b>Command line options</b>	<b>Description</b>
--help	Displays help for the create command
--name <name>	Name of the OS Installation Profile
--description <description>	Description of the OS Installation Profile (optional)
--customerID <id>	SA customer id to be used for the OS Installation Profile (you can find the id by looking at the properties of a customer in NGUI)
--platform <platform>	Platform to be used for the OS Installation Profile (use listPlatforms to get available platforms)
--MRLID <id>	Media Resource Locator ID to be used for the OS Installation Profile (use listMRLs option to get available MRLs in SA)
--installConfigFile <file path>	Path to the install configuration file to be used for the OS Installation Profile. The file is dependent on the chosen plat-

Command line options	Description
	form
-- buildCustomizationScriptID <id>	Build Customization Script id to be used for the OS Installation Profile (you can find the id by looking at the package properties in NGUI) (optional)
--winDiskPartitioningFile <file path>	Disk partitioning config file path to be used for the OS Installation Profile (used for Windows platforms) (optional)
--winDiskFormattingFile <file path>	Disk formatting config file path to be used for the OS Installation Profile (used for Windows platforms) (optional)
--winInstallDrive <Windows Drive Label>	Windows install drive to be used for the OS Installation Profile (used for Windows platforms) (optional, default is "C")
--winHWSignature <signature name> <file path>	Hardware signature and hardware signature file to be used for the OS Installation Profile (used for Windows platforms) (optional)

Install Profile Tool **update** specific options

Command line options	Description
--help	Displays help for the update command
--installProfileID <id>	The ID of the Install Profile that will be modified (you can find it in the properties of the OS Install Profile in NGUI)
--name <name>	New name for the OS Installation Profile (optional)
--description <description>	New description for the OS Installation Profile. Specifying an empty string ("") value will remove the existing Description (optional)

Command line options	Description
--customerID <id>	New SA customer id to be used for the OS Installation Profile Profile (you can find the id by looking at the properties of a customer in NGUI) (optional)
--MRLID <id>	New Media Resource Locator id to be used for the OS Installation Profile (optional)
--installConfigFile <file path>	Path to the new install configuration file to be used for the OS Installation Profile. The file is dependent on the chosen platform (optional)
--buildCustomizationScriptID <id>	New Build Customization Script id to be used for the OS Installation Profile. Specifying a 0 value will remove the existing script (you can find the id by looking at the package properties in NGUI) (optional)
--winDiskPartitioningFile <file path>	New disk partitioning config file path to be used for the OS Installation Profile (used for Windows platforms). Specifying an empty string ("") value will remove the existing script (optional)
--winDiskFormattingFile <file path>	New disk formatting config file path to be used for the OS Installation Profile (used for Windows platforms). Specifying an empty string ("") value will remove the existing script (optional)
--winHWSignature <signature name> <file path>	New hardware signature and hardware signature file to be used for the OS Installation Profile (used for Windows platforms). Specifying an empty string ("") value for the file path will remove the existing script for the specified hardware signature (optional)

Install Profile Tool **remove** specific options

Command line options	Description
--help	Displays help for the remove command
--installProfileID <id>	The ID of the Install Profile that will be removed (you can find it in the properties of the OS Install Profile in NGUI)

## Defining an OS Installation Profile — Linux/UNIX

To define a Linux/UNIX OS Installation Profile, use the `install_profile` script. Example:

```
/opt/opsware/mm_wordbot/util/install_profile create --user myUser
--customerID 0 --MRLID 74870001 --installConfigFile /root/rhel.cfg --platform 'Red Hat Enterprise Linux Server 6 X86_64' --name "rhel6 profile" -description "Install Profile description" -buildCustomizationScriptID 64870001
```

- 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”.
  - Platform:** (*Required*) Specify the platform of the Linux/UNIX operating.
  - Installation Media:** (*Required*) Specify the MRL ID for the Linux/UNIX operating system (use the `listMRLs` option from the `install_profile` script to view available MRLs).
  - See [Creating Media Resource Locators \(MRLs\)](#) for more information.
  - Build Customization Script:** (*Optional*) Specify the Build Customization Script ID to use for this installation profile. See [Using Build Customization Scripts](#)

The method by which 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 be used by OS Provisioning.

The file that you choose can have any file name, however, OS Provisioning renames the file with the file name required by the vendor installation program.

## Defining an OS Installation Profile — Windows

To define a Windows OS Installation Profile, use the `install_profile` script. Example:

```

/opt/opsware/mm_wordbot/util/install_profile create --user myUser
--customerID 0 --MRLID 74870001 --installConfigFile /root/un-
attend.xml --platform ' Windows 2008' --name "win 2008 profile" -
description "Install Profile description" -buildCus-
tomizationScriptID 64870001 -winDiskPartitioningFile /root/win-
part.txt -winDiskFormattingFile /root/winformat.txt -
winInstallDrive D -winHWSignature "VMWARE, INC. VMWARE VIRTUAL
PLATFORM" /root/hwsig.xml

```

1. Define 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.
- **Platform: (Required)** Specify the platform of the Windows operating system. 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.
- **OS Media: (Required)** Specify the MRL ID for the Windows operating system. See [Creating Media Resource Locators \(MRLs\)](#) for more information.
- **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. Default script is:
 

```

rescan
select disk 0
clean
create partition primary
activeassign letter=C

```
- **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. Default script is:
 

```

format.com C: /FS:NTFS /Q /Y /V:

```
- **Build Customization: (Optional)** Specify the ID of 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. See [Creating Build Customization Scripts](#) for more information.
- **Response File: (Required)** Select a Windows response file to upload into the OS Installation Profile. This can be an `unattend.xml` for unattended Windows



installations or a `sysprep.inf` type file for image type Windows installations.

- **Hardware Signatures:** (*Optional*) Define the hardware that the operating system supports.

### 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.

**Note:** The use of Hardware Signatures files is not required for Oracle Solaris/Sun SPARC or Red Hat Linux operating systems because Solaris/SPARC 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 Client. See [Defining Custom Attributes](#). For information on how to set custom attributes for software policies, see [Adding Custom Attributes to OS Installation Profile \(SA Client\)](#).
- Specifying custom disk partitioning and custom drive formatting (For Windows servers).

## Changing the OS Installation Profile Properties

To change the properties for an OS Installation Profile use the `install_profile` script. See the Install Profile Tool **update** specific options table.

## Modifying the OS Installation Profile Packages

With the release of SA 10.23, 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/SPARC 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](#) and [Creating OS Sequences](#) for more information.

## 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 Client.

To view the history of changes to an OS Installation Profile in the SA Client, perform the following steps:

1. Launch the SA Client: **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

**Note:** 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, use the `install_profile` script. See the **Install Profile Tool update specific options** table in [Import Media Tool Syntax and Options](#).

## Creating Build Customization Scripts

This section discusses the following topics:

- [Using Build Customization Scripts](#)
- [Solaris Build Customization Scripts](#)
- [Linux Build Customization Scripts](#)
- [Windows Build Customization Scripts](#)

## 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 By Folder view 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/SPARC 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](#) 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 6** details the exact steps that occur when you provision an installation client with Solaris.

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

Table 6. Sun Solaris Build Process

Phase	Build Process Steps
Pre-installation	<ol style="list-style-type: none"> <li>1. A user boots the installation client over the network by entering the following command in a console attached to the server:  <code>boot net:dhcp - install</code></li> <li>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.</li> <li>3. The OS Build Agent registers with the OS Build Manager.</li> <li>4. The Solaris build script probes the hardware configuration of the installation client and registers it with SA.</li> </ol>
Phase One	<ol style="list-style-type: none"> <li>5. In the SA Client, a user chooses to install an operating system on an available installation client.</li> </ol>

Phase	Build Process Steps
	<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:</p> <pre data-bbox="695 709 1156 741">/sbin/sh run Pre-JumpStart</pre> <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 <code>pfinstall</code> 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 SA Client.</p> <p>12. The JumpStart installer <code>pfinstall</code> 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:</p> <pre data-bbox="695 1501 1175 1533">/sbin/sh run Post-JumpStart</pre> <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:</p> <pre data-bbox="695 1843 1084 1875">/sbin/sh run Pre-Agent</pre>

Phase	Build Process Steps
	<p>17. The Solaris build script installs the SA Agent.</p> <p>18. The Solaris build script executes the build customization script:</p> <pre data-bbox="695 436 1101 468">/sbin/sh run Post-Agent</pre> <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 Client output details of the `run` job window.

#### 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 7** 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 7. Linux Build Process

Phase	Build Process Steps
Pre-installation	<ol style="list-style-type: none"> <li>1. A user boots the installation client from PXE or the Linux Boot CD ROM.</li> <li>2. The installation client loads a standard Red Hat boot image and mounts the second stage image specified by the kernel parameters.   <b>Note:</b> 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.</li> <li>3. The OS Build Agent registers with the Build Manager.</li> <li>4. The Linux build script probes the hardware configuration of the installation client and registers it with SA.</li> </ol>
Phase One	<ol style="list-style-type: none"> <li>5. In the SA Client, a user selects the target version of Linux to install on the installation client.</li> <li>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.</li> <li>7. The Linux build script copies GRUB or ELILO onto the partition and installs it into the MBR.</li> <li>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.</li> <li>9. If the Custom Attribute <code>kernel_arguments</code> is set for the OS Installation Profile, these kernel arguments are appended.</li> <li>10. The OS Build Agent exits and the server reboots.</li> </ol>
Phase Two	<ol style="list-style-type: none"> <li>11. The target boot image loads and runs the OS Build</li> </ol>

Phase	Build Process Steps
	<p>Agent.</p> <ol style="list-style-type: none"> <li>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.</li> <li>13. The Linux build script writes the configuration file defined by the MRL to the disk.</li> <li>14. If it exists, the Linux build script runs the build customization script.</li> <li>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.</li> <li>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.</li> <li>17. When the operating system installation completes, Anaconda reboots the installation client, which boots from the newly installed operating system.</li> </ol>
Phase Three	<ol style="list-style-type: none"> <li>18. On entering multi-user mode, the OS Build Agent is invoked and contacts the OS Build Manager.</li> <li>19. The Linux build script installs the SA Agent.</li> <li>20. The Linux build script exits.</li> </ol> <p>The operating system installation section of provisioning is complete.</p>

#### Sample Solaris Build Customization Script

```
#!/sbin/sh

pre_jumpstart() {
#
# strip any partitioning information out of profile, and
# replace it with keywords to use default partitioning, but
# to size swap equal to the amount of physical RAM
#
cat $SI_PROFILE | grep -v partitioning | grep -v filesys > /tmp/profile.$$
echo "partitioning default" >> /tmp/profile.$$
```

```

echo "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

```

### 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](#).

### 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)

**Note:** In order to perform PXE booting of a VMware ESX Windows Server 2003 x86 or x86\_64 VM using WinPE, the minimum required RAM is 512MB (higher than the VMware recommended RAM minimum).

**Table 8** details the steps that occur when you provision an installation client with Windows WinPE.

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

Table 8. Microsoft Windows Build Process(WinPE)

Phase	Build Process Steps
Pre-installation	<ol style="list-style-type: none"> <li>1. A user boots an installation client over the network by using a PXE network bootstrap program or by using the WinPE.</li> <li>2. The user can install either WinPE x86 32 bit or WinPE x64 64 bit pre-installation environment.</li> <li>3. PXE boots the Windows OS Build Agent over the network.</li> </ol> <p>When using the WinPE pre-installation environment, you will not be prompted to create a disk partition.</p> <ol style="list-style-type: none"> <li>4. The OS Build Agent collects pertinent hardware information and registers the information with SA.</li> </ol> <p>The server is ready to be provisioned and is available for selection from the Unprovisioned Servers in the SA Client.</p>
Phase One	<ol style="list-style-type: none"> <li>5. The user selects a Windows server from the Unprovisioned Servers list in the SA Client and runs an OS Sequence on the server.</li> <li>6. The Windows build script mounts the Windows installation media as indicated by the Media Resource Location (MRL).</li> <li>7. The Windows build script initiates a Windows unattended setup.</li> <li>8. The Windows build script waits for a Windows unattended setup to complete and Windows to boot for the first time.</li> </ol>
Phase Two	<ol style="list-style-type: none"> <li>9. Windows boots for the first time.</li> <li>10. If a build customization script was specified in the OS Installation Profile, it is executed by the Windows build script.</li> <li>11. The Windows build script installs the Agent.</li> </ol> <p>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. 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 10 and 11](#)
- [Custom Attributes for Linux or VMware ESX](#)
- [Custom Attributes for Microsoft Windows](#)
- [Adding Custom Attributes to OS Installation Profile \(SA Client\)](#)

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

The SA Client provides 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 \(SA Client\)](#)

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 \(SA Client\)](#) for specific steps required to set custom attributes for an OS Installation Profile.

### Custom Attributes for Sun Solaris 10 and 11

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 Client.

The custom attributes correspond to the equivalent keywords in the `sysidcfg` file. See **Table 9**.

Table 9. Sun Solaris 10 and 11 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><b>Example Value:</b> <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><b>Example Value:</b> 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><b>Example Value:</b> <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><b>Example Value:</b> 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><b>Example Value:</b> Any value in the <code>/usr/share/lib/zoneinfo</code> directory on a solaris server.</p>

Keyword	Description
<code>system_locale</code>	<p>Sets the language for the configuration of the installation client (sets LANG in <code>/etc/default/init</code>). Valid locale values are installed in <code>/usr/lib/locale</code>. If you set this attribute, you should also use the locale keyword in the operating system profile so that the appropriate locale is installed.</p> <p>By default, the value for this keyword is <code>system_locale=C</code>.</p> <p><b>Example Value:</b> "C", "en_US.UTF-8", "ja_JP.UTF-8".</p> <p>See <a href="http://developers.sun.com/dev/gadc/faq/locale.html">http://developers.sun.com/dev/gadc/faq/locale.html</a></p>
<code>required_patches</code>	No longer supported.
<code>nfsv4_domain</code>	<p>Sets the system's default NFS version 4 domain name. This value is substituted into <code>/etc/default/nfs</code> 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><b>Example Value:</b> company.com</p>
<code>mrl</code>	(Solaris 11) The MRL created by the <code>import_media</code> script. This is useful when you want to use an external repository.
<code>http_proxy</code>	(Solaris 11) Specify when you use HTTP repositories.
<code>enable_root_user</code>	<p>(Solaris 11) Solaris 11 allows root to be defined as a role in the <code>sc.xml</code> file.</p> <p>Presence of this custom attribute will trigger the user creation, even if 'no' value is set.</p>
<code>hostname</code>	Specifies the hostname of the machine.

### 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).

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

Table 10. Linux or VMware ESX Custom Attributes

Keyword	Description
boot_disk	<p>Values: A raw device name without <code>/dev/</code> such as <code>"sda"</code>, <code>"hdc"</code>, <code>"cciss/c0d1"</code></p>
boot_kernel	<p>Values: <b>"rhel30"</b>, <code>"rhel40"</code>, <code>"rhel50"</code>, <code>"reh160"</code>, <code>"rhel3ia"</code>, <code>"rhel4ia"</code>, <code>"rhel5ia"</code></p> <p><b>Note:</b> This custom attribute is used only for reprovisioning. The value of this custom attribute specifies the type of kernel the server boots to during reprovisioning.</p>
hpsa_netconfig	Created after using non-DHCP to boot the target server into the Unprovisioned Servers list.
kernel_arguments	<p>Values: <code>"noapci"</code>, <code>"root=LABEL=/"</code>, <code>"quiet"</code>, <code>"splash"</code></p>
ksdevice (Linux pxe boot)	<p>Values: MAC address of the NIC</p> <p><b>Note:</b> 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:</p> <pre>kernel_arguments      =ksdevice=mac address</pre> <pre>ksdevice              mac address</pre> <p>When powering on and PXE booting a device, you do not need to specify the kickstart device.</p>
ksdevice (linux5, linux6)	<p>Values: <code>bootif</code></p> <p>Default: <code>ksdevice=bootif</code></p> <p>Use for all Linux PXE types (including <code>linux5</code>, <code>linux6</code>) to prevent prompting for the Kickstart device when boot-</p>

Keyword	Description
	ing a multiple NIC server into the Unprovisioned Server pool.
nfs_opts	<p>Use <code>--opts</code> to specify NFS options in the <code>ks.cfg</code>. (when provisioning Red Hat Enterprise Linux 5 or later).</p> <p>For example:</p> <pre>nfs --server &lt;Server IP&gt; --dir &lt;media director&gt; --opts &lt;nfs options&gt;</pre> <p>For example, to contain a comma delimited set of values the same as the NFS values allowed in <code>/etc/fstab</code>, create a custom attribute <code>nfs_opts</code> with the value <code>"rsize=32768, wsize=32768"</code>.</p>
timeout	<p>Values: the number of minutes to wait for Linux provisioning to complete before timing out.</p> <p>Default: 30 minutes</p> <p>If Linux provisioning fails because the job takes too long to complete, you can specify a longer timeout period.</p>

### 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 10 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.

**Note:** 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 Opware
#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 \(SA Client\)](#).

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: Windows Custom Attributes for OS Provisioning**


<b>Keyword</b>	<b>Corresponding unattend.txt Attribute</b>	<b>Description</b>
<code>AdminPassword</code>	<code>[GuiUnattended] /AdminPassword</code>	This option sets the Administrator password for the Admin account.
<code>AGENT_INSTALL_DELAY</code>		Allows you to introduce a delay after provisioning a system that allows the build scripts to wait before starting to install the agent.  Default: 30 seconds
<code>argstring</code>	None	String value that is used to compose the command line arguments for the Agent installer.
<code>auto_partition</code>		Used by consoleless to indicate that instead of requiring interactive user confirmation before partitioning the disk, partition the disk automatically.
<code>ComputerName</code>	<code>[UserData]/ComputerName</code>	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.



<b>Keyword</b>	<b>Corresponding unattend.txt Attribute</b>	<b>Description</b>
hpsa_netconfig	None	Created after using non-DHCP to boot the target server into the Unprovisioned Servers list.
imageexec	None	Command to apply legacy image-based provisioning image. This supports traditional imaging tools such as Symantec Ghost™. However, using the built in support for WIM images is strongly encouraged.
imagefile	None	Path to a server image file. This supports traditional imaging tools such as Symantec Ghost™. However, using the built in support for WIM images is strongly encouraged.
imageshare	None	Share with image file to install. This supports traditional imaging tools such as Symantec Ghost™. However, using the built in support for WIM images is strongly encouraged.
ProductKey	[UserData]/ProductKey	This value is not validated by SA.
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.

### **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 the following method: From **Start > All Programs > SA Client**.
2. From inside the SA 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 want 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 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.

**Requirement:** 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 *SA User 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.

**Note:** 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.

## **The Manage Boot Clients 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 with either an OS Build Plan or an OS Sequence 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, 3 or 4 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 Build Plan or 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 `/Opsware/Tools/OS Provisioning/Manage Boot Clients` folder).

For iLO2, 3 or 4 integration, the user must have Manage iLo and Execute iLo operations permissions.

## Installation

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

The following four APXs are installed for MBC:

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

## Using the Manage Boot Clients 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 typically changes the default PXE menu choice when the server PXE boots, so that you are not required to choose a PXE image from the console of the server that is booting up. MBC also allows you to associate an OS Build Plan or an OS Sequence with the server record so that, when the server registers as an unprovisioned server with SA, a provisioning job starts automatically.

## Running an MBC APX

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/](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 OGS 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>buildplan_id</code>	If specified, will invoke an OS Build Plan installation as the user using MBC as soon as the server is added to the Server Pool.  <b>Note:</b> <code>buildplan_id</code> is stored as a custom attribute on the server. This custom attribute is removed from the server record when a Build Plan is started on the server.
<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> , <code>linux6</code> or <code>linux6-x64</code> when using an OS Build Plan, or <code>winpe32</code> , <code>winpe64</code> , <code>linux5</code> or <code>linux6</code> when using an OS Sequence.). This copies the configuration file <code>/opt/opsware/boot/tftpboot/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.  <b>Note:</b> <code>sequence_id</code> 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 ID association for the server. This is neces-



Parameter	Description
	sary 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).
ilo.*	See <a href="#">iLO Integration</a> .
dhcpcleanup	Retrieves the DHCP configuration or deletes a DHCP entry by MAC address. Options are: <ul style="list-style-type: none"> <li>- help (-h): Displays online help</li> <li>- action (-a): Options are: <ul style="list-style-type: none"> <li>get: Retrieves the DHCP configuration</li> <li>delete: Deletes a DHCP server(s) from the configuration. You must also specify either the: <ul style="list-style-type: none"> <li>The MAC addresses or</li> <li>--facility (-f)=FACILITYNAME: specify which facility's DHCP servers to operate on.</li> <li>--macs (-m)=MACS: a comma-separated list of MAC addresses to remove from the DHCP configuration.</li> <li>--outputdir (-o)=OUTPUTDIR: when specified, MBC saves progress and results information in the specified directory.</li> </ul> </li> </ul> </li> </ul>

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=linux6,
buildplan_id=2110061
00:0c:29:f9:12:f3,hostname=testvm2,pxe_image=winpe32
00:0c:29:0d:ab:b4,pxe_image=winpe64, buildplan_id=2110061
```

These CSV entries would cause MBC to create three Planned Server records and set them up to boot to the `linux6`, `winpe32`, and `winpe64` PXE images, respectively. The servers processed by the first and third CSV entries will also have an OS Build Plan 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 would be similar to `dhcp-client-00:0c:29:0d:ab:b4`.

**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=  
e=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
<code>dhcp.hostname</code>	Specifies the MAC address for hostname(s) that are authorized for DHCP leases.
<code>dhcp.ip</code>	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, 3 and 4 (iLO2, iLO3, iLO4) Standards. 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
<code>ilo.hostname</code>	Hostname or IP address for the iLO. This must be accessible from the <code>hub</code> server. This value is stored as a custom attribute by MBC.
<code>ilo.username</code>	Username to use to authenticate to the iLO. This value is stored as a custom attribute by MBC.
<code>ilo.password</code>	Password used to authenticate to the iLO. This value is not stored as a custom attribute by MBC.
<code>ilo.reboot_if_on</code>	<b>Default:</b> 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.host-
name=danube,ilo.hostname=10.128.32.102, ilo.user-
name=Administrator,ilo.password=adminpass,
ilo.reboot_if_on=1
```

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

If you plan to use SA 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 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 SA 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 14**:

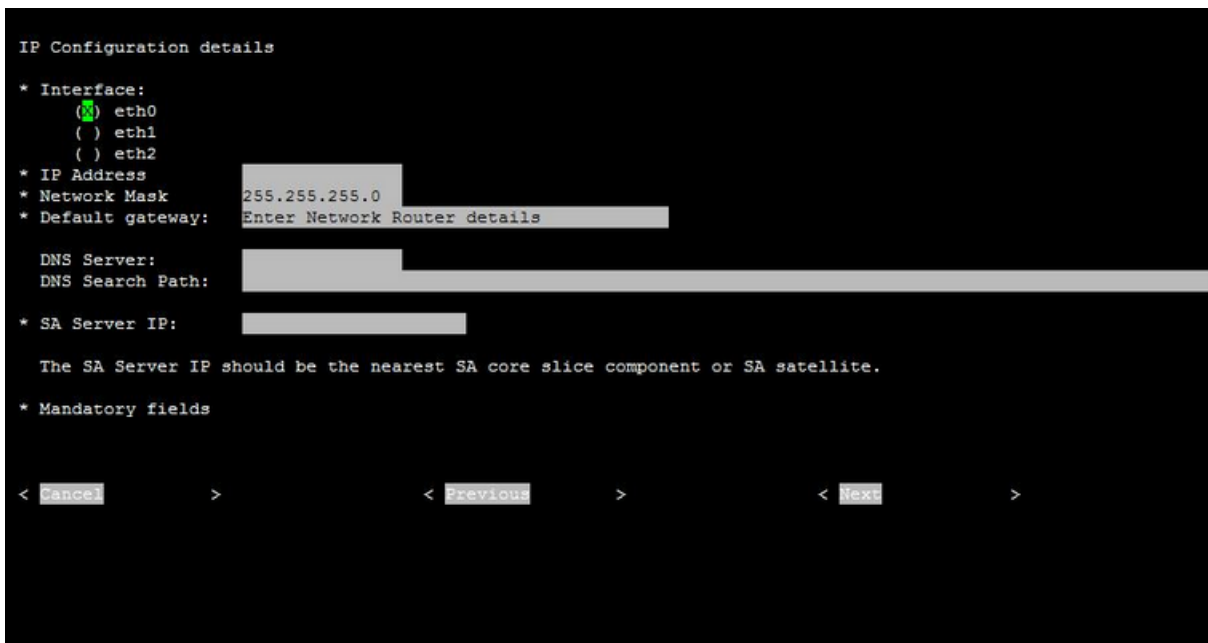
**Figure 14. Red Hat Linux Boot Screen**



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 15**:

**Figure 15. Red Hat Linux Network Configuration Dialog**



You can manually configure the following fields:

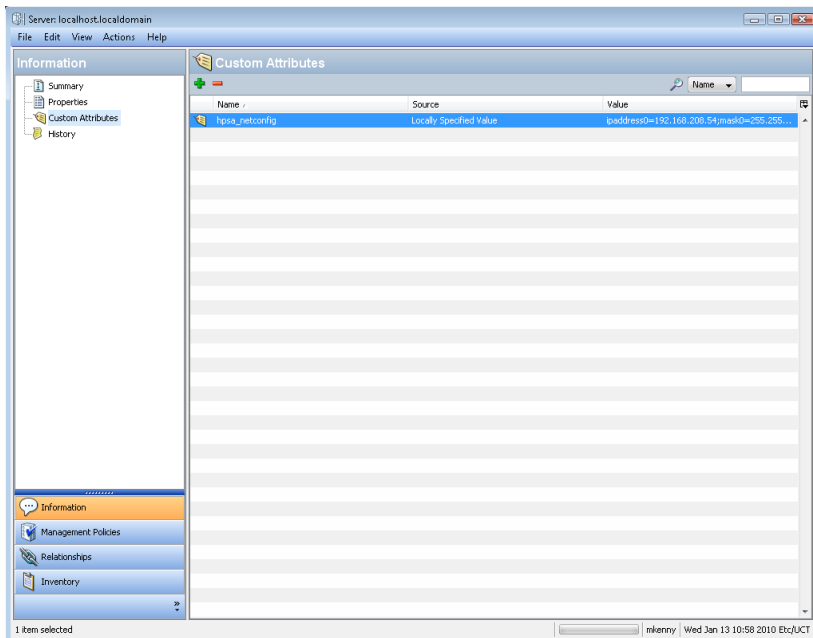
- Interface: the NIC to be used
- IP Address: static IP address for the server being provisioned
- Netmask: netmask for the server being provisioned
- Default gateway: Gateway IP address the server being provisioned should use (network level IP router)
- DNS Server: the IP address the server being provisioned 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 SA 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 16**:

**Figure 16. hpsa\_netconfig Custom Attribute in Server Record**



## Booting a Red Hat Enterprise Linux Itanium 64-bit Server in a Non-DHCP Environment Using Elilo Boot

If you plan to use SA 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 > Opsware > 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 below:

```
HP SA Linux Boot CD (<version>):
Enter the appropriate Linux service OS
at the 'Elilo boot:' prompt.
```

linux5 - RHEL 5.7 based Linux service OS

linux5-txt - RHEL 5.7 based Linux service OS for serial consoles

ELILO boot:

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 17. Red Hat Linux Itanium 64-bit Network Configuration Dialog**



**Note:** 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:

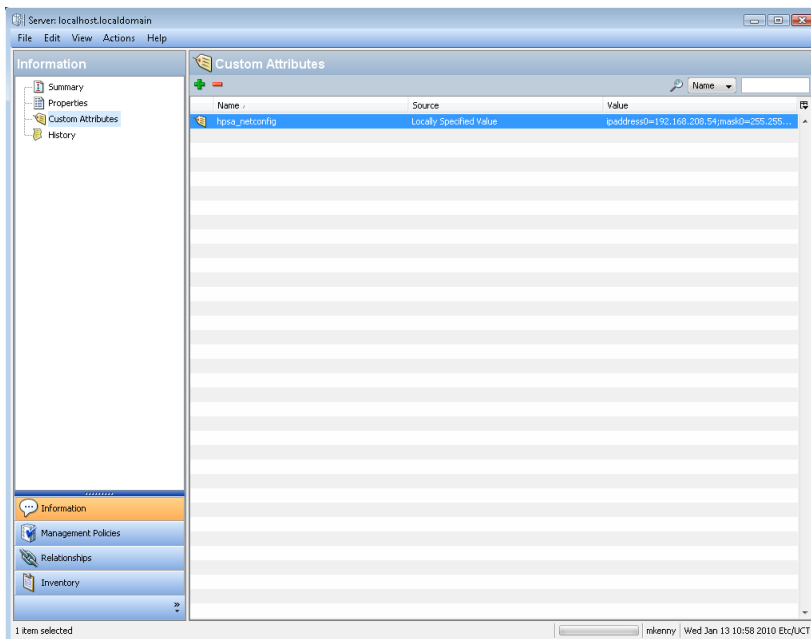
- Interface: the NIC to be used
- IP Address: static IP address for the server being provisioned
- Netmask: netmask for the server being provisioned
- Gateway: Gateway IP address the server being provisioned should use (network level IP)
- DNS Suffix: the fully qualified DNS suffix the server being provisioned 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 SA 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 18**:

**Figure 18. hpsa\_netconfig Custom Attribute in Server Record**



### Booting a Windows Server in a Non-DHCP Environment

If you plan to use SA 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.

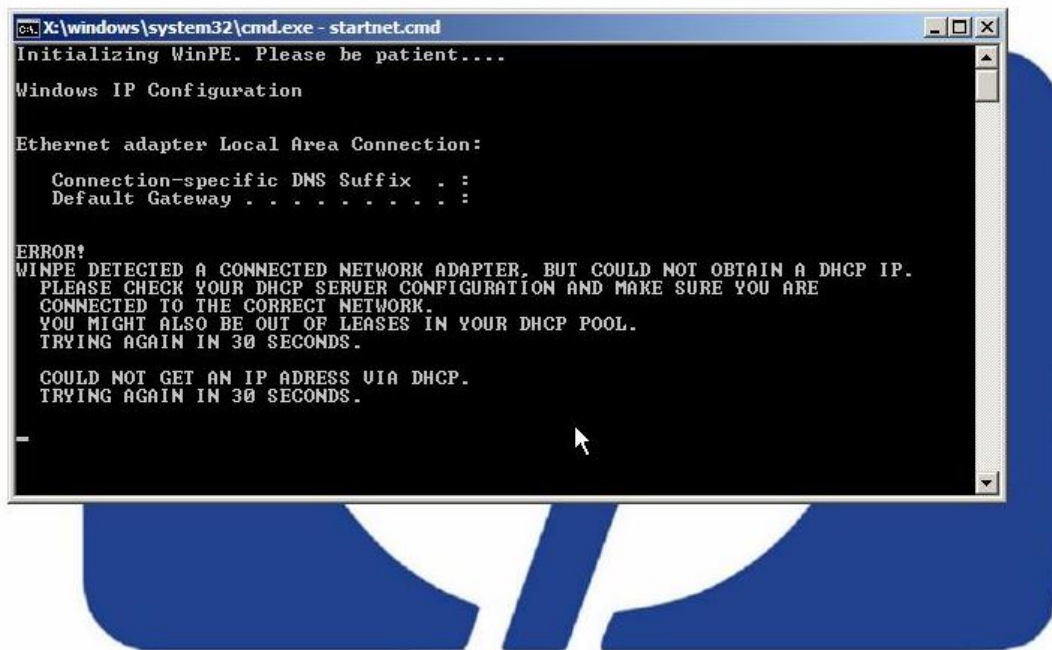
This section provides details for provisioning in a non-DHCP environment.



### 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 19**.

**Figure 19. WinPE Console Display when DHCP Server Not Found**



WinPE x64 (64 bit)  
HP boot image version: 37.0.0.0.9

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 20**:

**Figure 20. WinPE Network Configuration**

Interface\* Intel(R) PRO/1000 MT Network Connectio

MAC Address: 00:0C:29:D3:72:DE

Static IP  DHCP

IP Address: 169.254.226.67

Netmask: 255.255.0.0

Gateway: 0.0.0.0

DNS Server: . . . .

DNS Suffix:

Server Automation Agent Gateway

Hostname / IP:

Port:\* 8017

OK Cancel

Select the correct Interface and specify the Static IP.

You can manually configure the following fields:

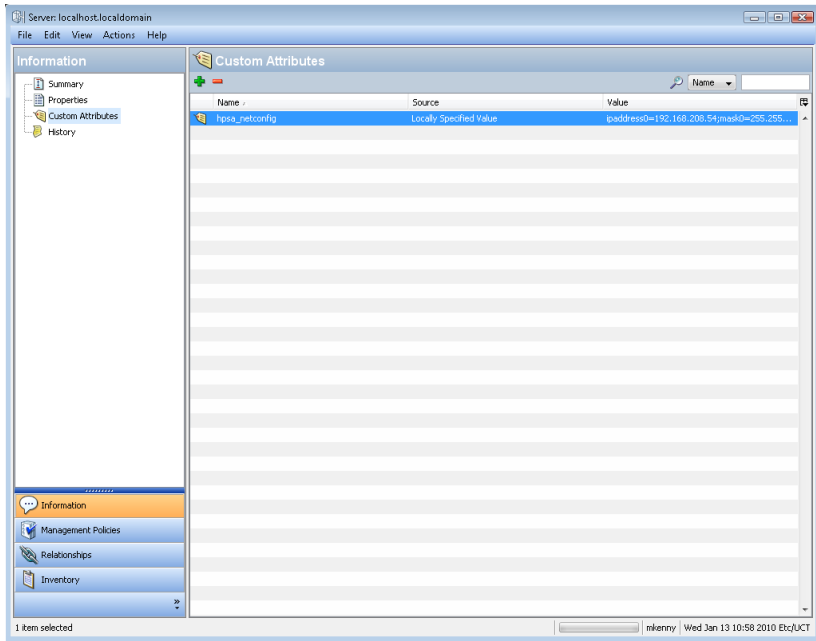
- IP Address: static IP address for the server being provisioned
- Subnet: Subnet mask for the server being provisioned
- Gateway: Gateway IP address the server being provisioned should use (network level IP router)
- DNS Server: the IP address the server being provisioned should use
- DNS Suffix: the 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 21**:

**Figure 21. hpsa\_netconfig Custom Attribute in Server Record**



# 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.

### 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"
}

```

If you use HPUX OS Provisioning, the following steps are required on the SA Core with Ignite installed:

1. Edit the `/etc/exports` file
2. Change the line the following line:

```

/var/opt/ignite/clients *(ro,no_root_squash,async)
to
/var/opt/ignite/clients *(rw,no_root_squash,async)

```

3. Run `"exportfs -a"`

## 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 > Opsware > 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 > Opsware > 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 > Opsware > 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.

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

### 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

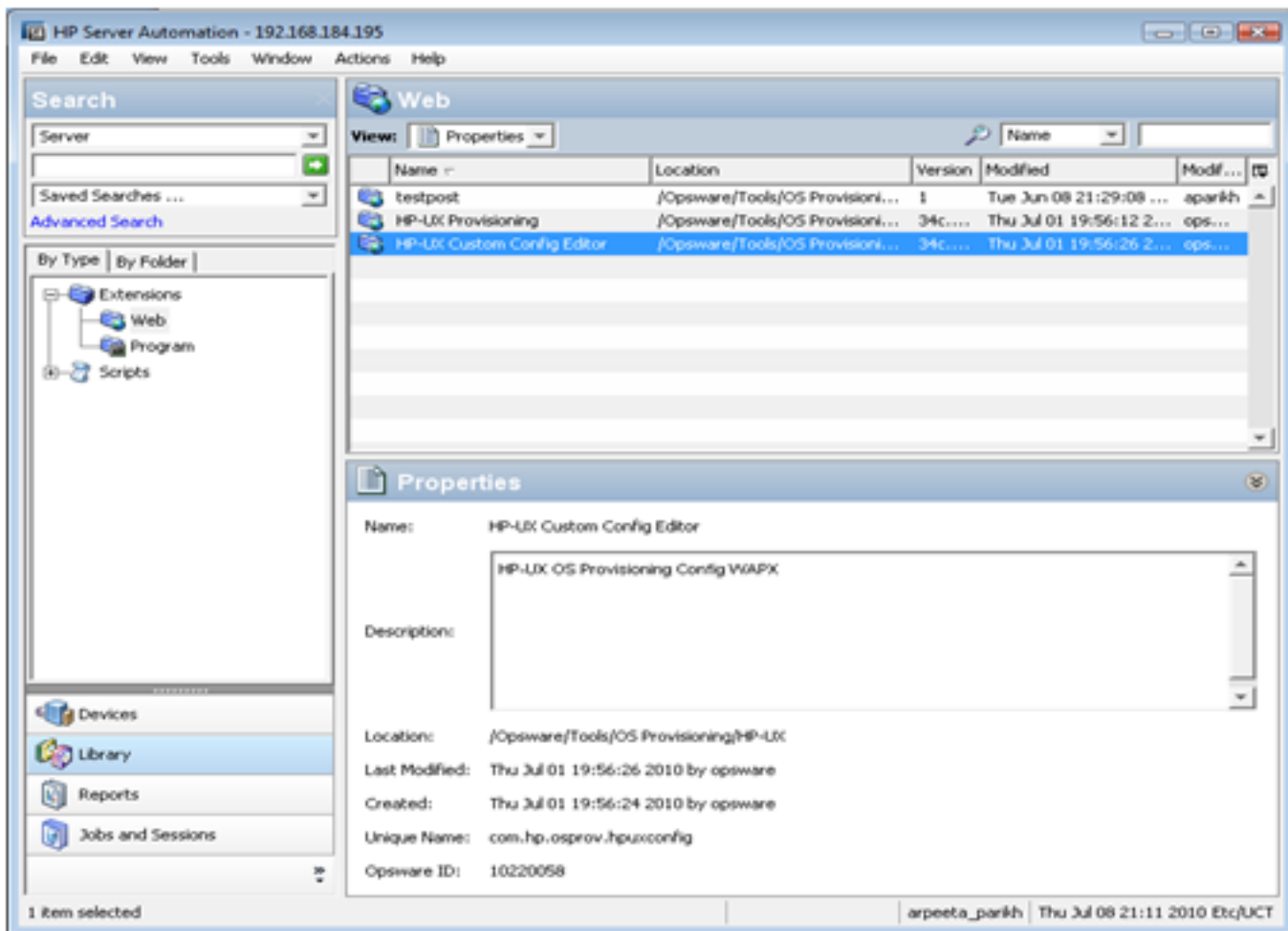
To access the HP-UX Custom Configuration Editor APX, perform these tasks:

1. In the SA 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`.

### 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.

### HP-UX Custom Configuration Editor APX – List Existing Configurations



HP-UX Configurations in SA

Create/Delete HP-UX Configurations

Name	Customer	HP-UX Platform	Ignite Server	Base Config
testing	Not Assigned	HP-UX 11.31	tomato2.tomato.qa.opsware.co	HP-UX B.11.31_vPar Default
LVM 11.31 vPar Satellite	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
11.23 vPar for Slave J3	Not Assigned	HP-UX 11.23	jade3.jade.qa.opsware.com	HP-UX B.11.23_vPar Default
11.23 Standalone	Not Assigned	HP-UX 11.23	tomato2.tomato.qa.opsware.co	HP-UX B.11.23 Standalone Default
11.23 Standalone Correcte	Not Assigned	HP-UX 11.23	tomato2.tomato.qa.opsware.co	HP-UX B.11.23 Standalone Default
11.31 standalone for mast	Not Assigned	HP-UX 11.31	tomato2.tomato.qa.opsware.co	HP-UX B.11.31_standalone Defaul
11.31 vPar Slave Core	Not Assigned	HP-UX 11.31	jade3.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
11.23 vPar on slave core	Not Assigned	HP-UX 11.23	jade3.jade.qa.opsware.com	HP-UX B.11.23_vPar Default
11.31 vPar for satellite	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
LVM 11.31 vPar for Satellite	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
LVM 11.31 vPar for n026	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
LVM-2 11.31 SA on n026	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_standalone Default
11.23 test sw policy	Not Assigned	HP-UX 11.23	tomato2.tomato.qa.opsware.co	HP-UX B.11.23 Standalone Default
LVM Test on Satellite	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
LVM test 3 for satellite	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
LVM FS Test	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
LVM Test 2	Not Assigned	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
11.23 vPar for multi target	Customer A	HP-UX 11.23	jade3.jade.qa.opsware.com	HP-UX B.11.23_vPar Default
validate 11.23 vPar on sla	Customer A	HP-UX 11.23	tomato2.tomato.qa.opsware.co	HP-UX B.11.23_vPar Default
validate 11.2 vPar on slavi	Customer A	HP-UX 11.23	jade3.jade.qa.opsware.com	HP-UX B.11.23_vPar Default
vPar for LVM on Satellite	Customer A	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default
LVM attributes on vPar k08	Customer A	HP-UX 11.31	jade4.jade.qa.opsware.com	HP-UX B.11.31_vPar Default

Refresh Create Delete

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.

### HP-UX Custom Configuration Editor – Create Custom Config

**Create an HP-UX Configuration**

**NOTE:** Please use Internet Explorer to run this APX to create the configuration

**Config Name**

**Config description**

**Customer**

**Platform**

**Base Configuration**

**Custom Config**

**Software Policies**

Related Software Policies	Selected Software Policies
HP-UX Provisioning Ignite Server	swapoff_B.11.31.0809.02_HP-UX_B.11.31_IA_PA
MallocNextGen_B.11.31.0903.02_HP-UX_B.11.31	MallocNextGen_B.11.31.0903.02_HP-UX_B.11.31_I
MD5Checksum_A.01.01.02_HP-UX_B.11.31_IA+	NumericUser_B.11.31.0809.03_HP-UX_B.11.31_IA
DRD_1131_WEB1002 Depot	
NumericUser_B.11.31.0809.03_HP-UX_B.11.31	
SwPkgBuilder_A.02.00.32.450_HP-UX_B.11.31	
swapoff_B.11.31.0809.02_HP-UX_B.11.31_IA_P	

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

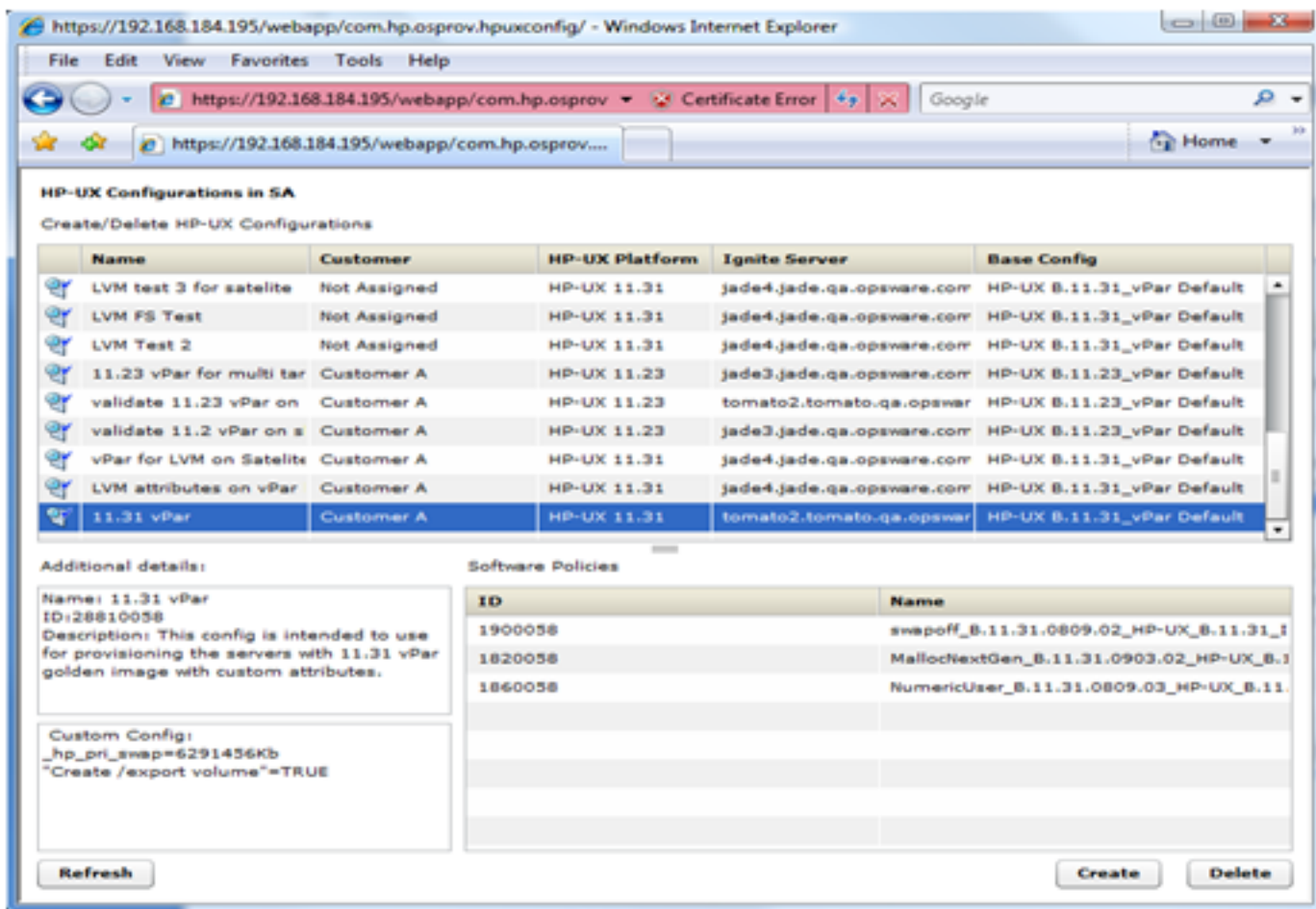
the policies to the selected software policies list to apply them on the server after provisioning is completed and the agent is installed.

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.

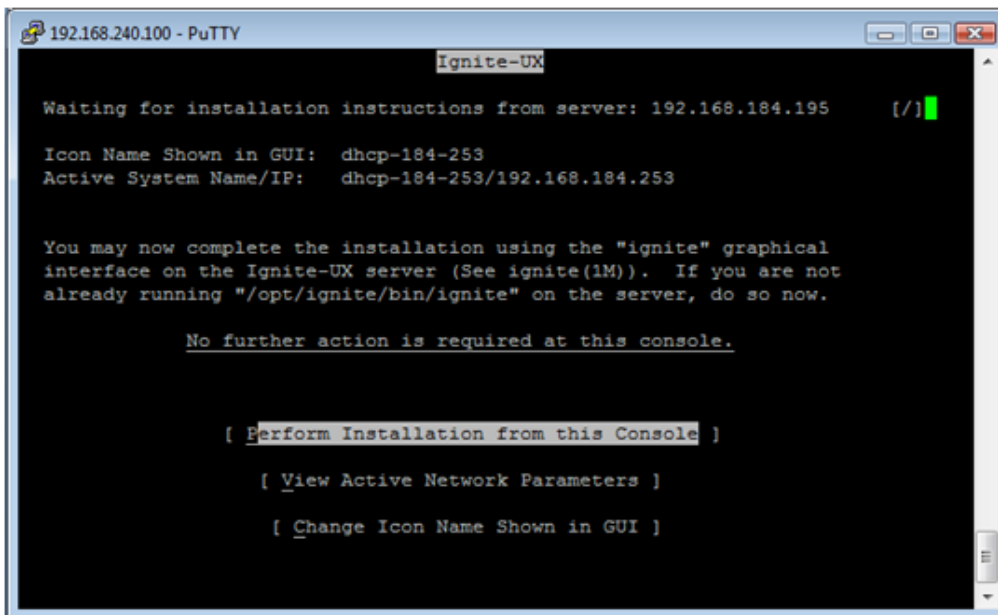
### Newly Created Custom Config Profile



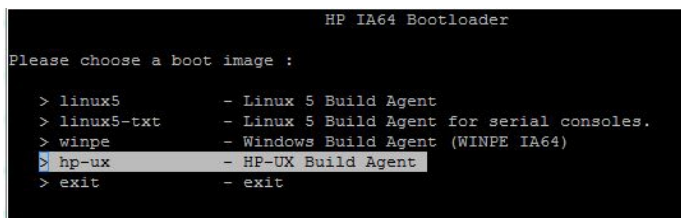
### Boot Target

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 26. Server Waiting to be Installed**



When you lanboot an HP-UX Client, a bootloader menu will provide an HP-UX option:



For more information, see:

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

## Provision the Target Servers

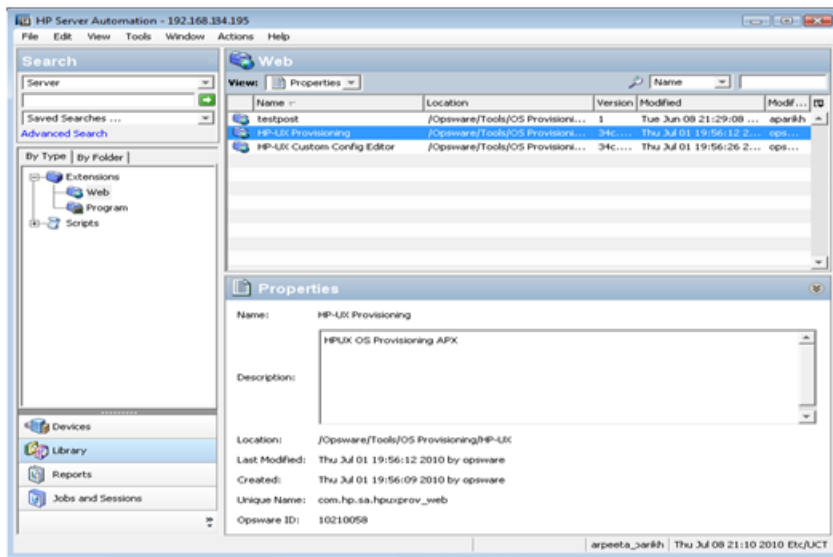
Once the custom configuration is created, it is listed on the HP-UX Provisioning APX on the SA Client. The target server waiting at the network boot prompt is listed under the unprovisioned servers list on the HP-UX Provisioning APX. The following section describes how to provision the targets.

### HP-UX Provisioning APX

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

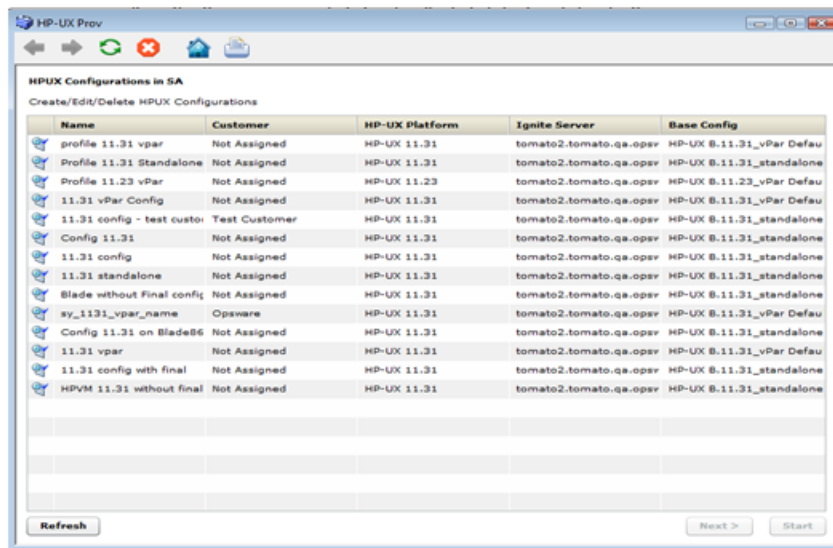
1. Log on to the SA Client.
2. Click the Library tab and select Extensions > Web > HP-UX Provisioning.

**Figure 27. HP-UX Provisioning APX**



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

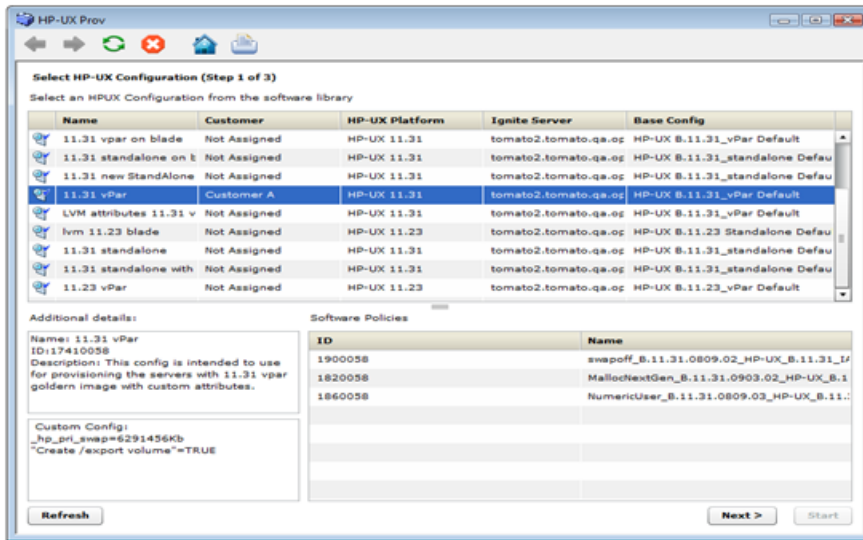
**Figure 28. HP-UX Provisioning APX – Listing of HP-UX Configurations**



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 29. HP-UX Provisioning APX – Select HP-UX Configuration**

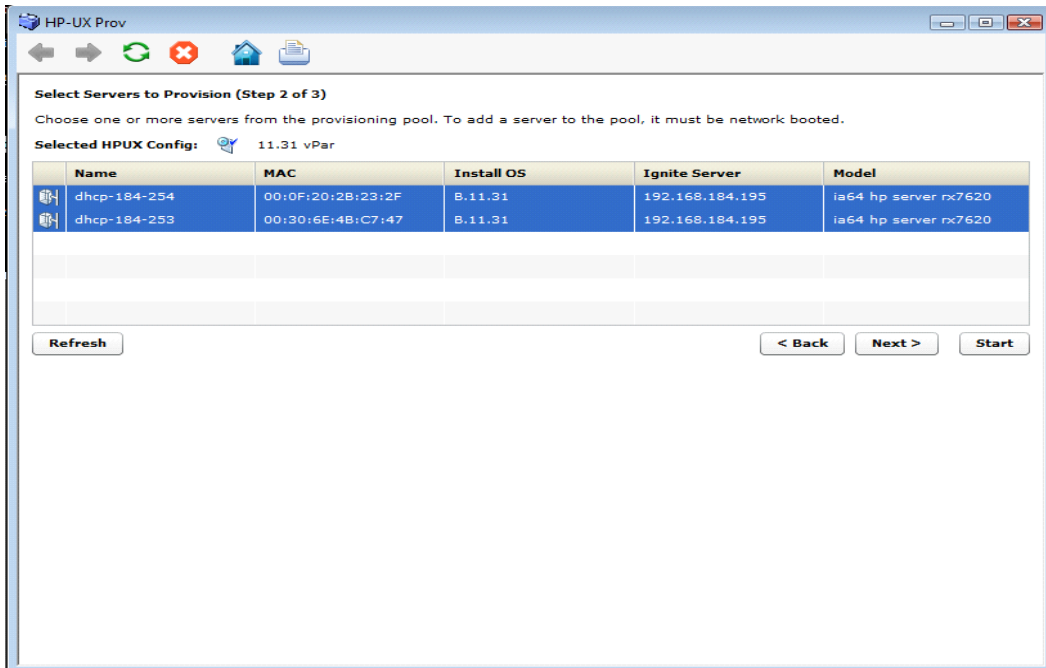


3. Click **Next** to select the unprovisioned servers.

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 30. Select Client to Provision Using the Config Selected Before**



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

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

**Figure 31. Set the Email Notification**

**Additional Job Options (Step 3 of 3)**  
 Optionally specify people to notify when the job is completed, and whether they should be notified if an error occurs. Also provide a ticket ID, if required.  
 Selected HP-UX Config:

**Email Notifications**

Email Address of Recipient	Success	Failure
parikh@hp.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
test@yahoo.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add Remove

**Ticket Integration**  
 Ticket ID:

< Back Start

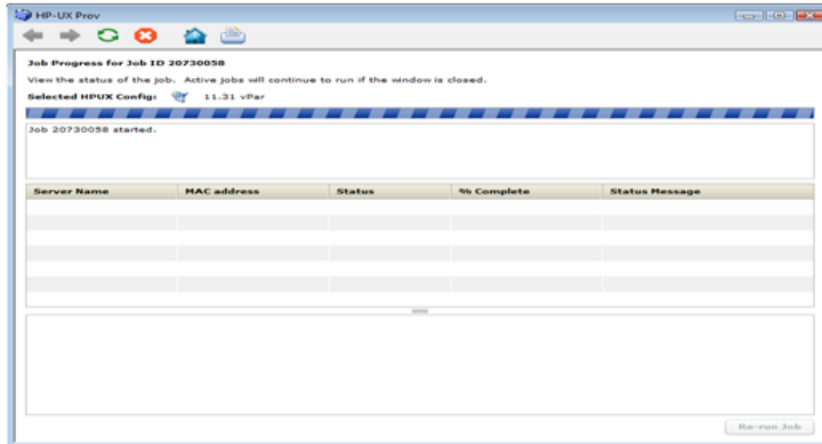
On Email Notification, by default, your email address (the user running the job) is displayed. To add additional email addresses, click **+Add**. Select the check boxes to receive notifications when job failure or success occurs. To remove an email address, select the address and click **Remove**.

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

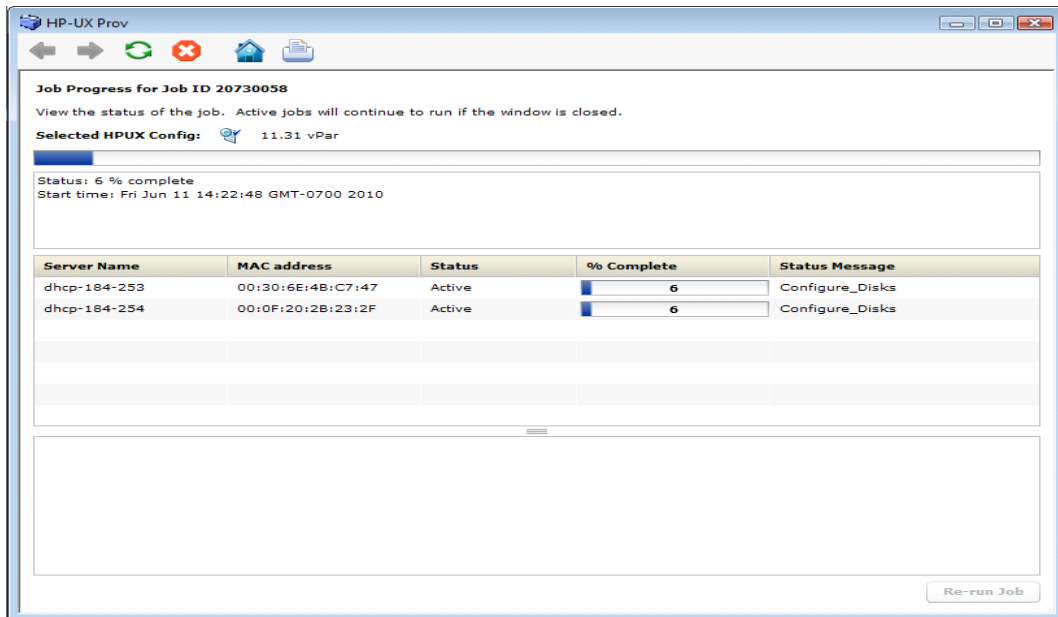
- When you click Start, the job is initiated for program APX and the Job ID is assigned to it.

**Figure 32. Initializing the Provisioning Job**





**Figure 33. Initializing the Provisioning Job**

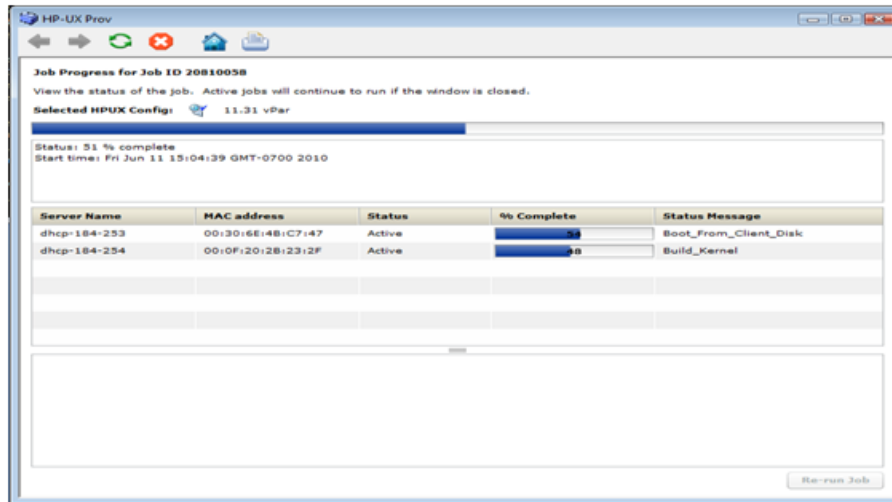


If the provisioning job was initiated successfully on the servers, you will see the following screen. This screen has a progress bar that is refreshed with updated progress status messages. The following status messages are updated during the provisioning job:

- 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

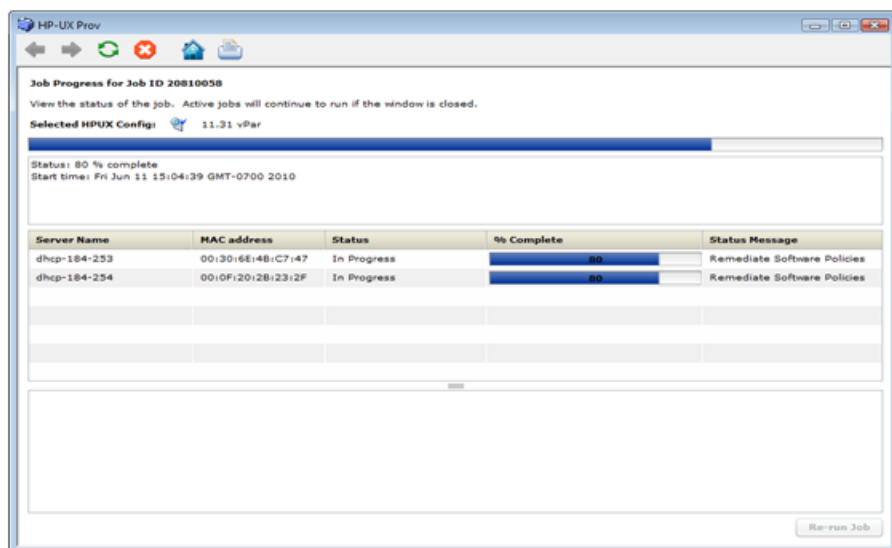
**Figure 34. Progress Bar with Status Message**



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.

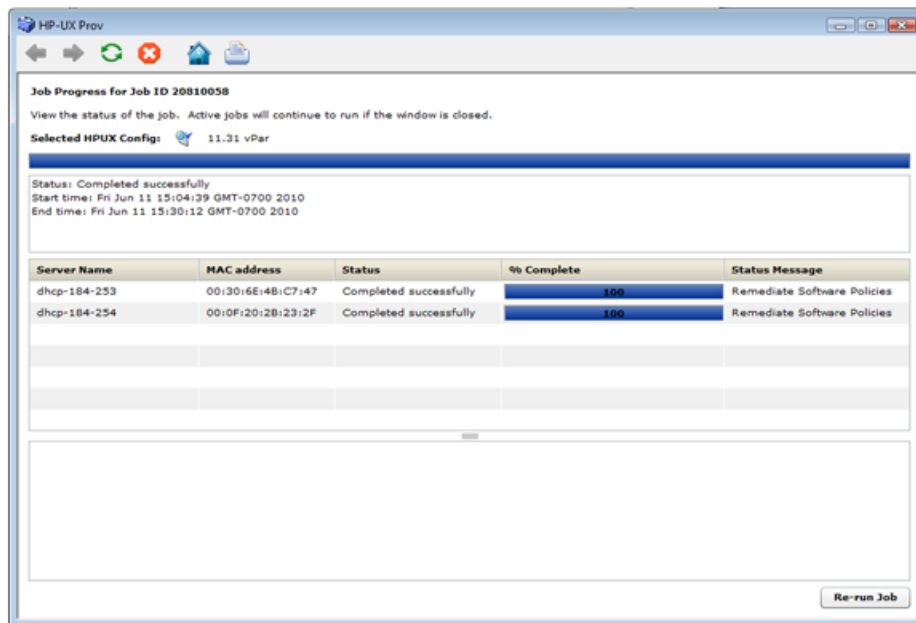
The progress bar for each of the servers being provisioned shows the percentage of provisioning complete with Server Name, MAC address, Status, and Status Message details. The % Complete and Status Message are updated along with the progress of the provisioning job.

**Figure 35. 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 36. Progress Bar with Job Completed Message**



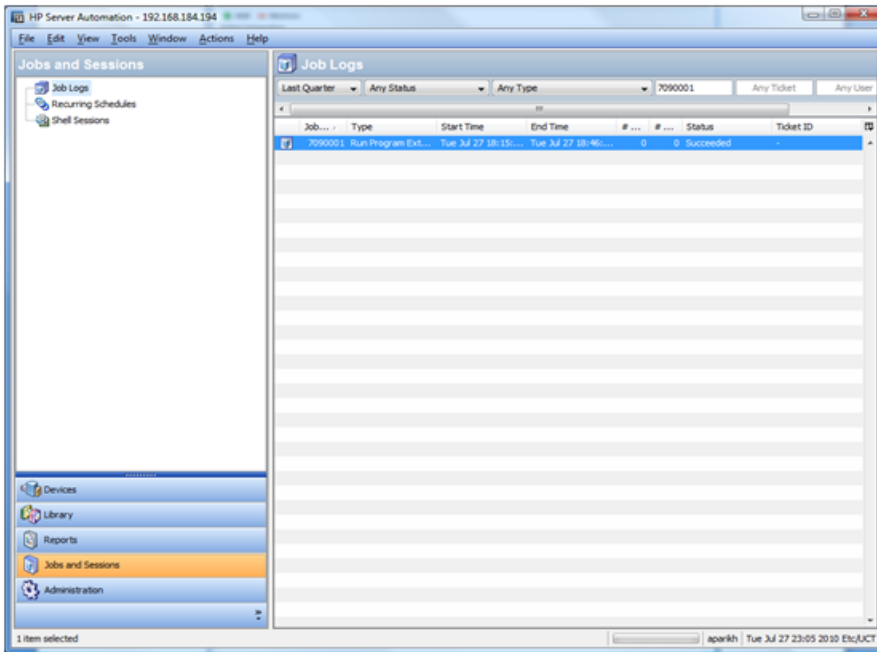
The provisioned servers are managed by SA. The servers are listed under Devices > All Managed Servers. Select the server, and then select Go to View > Properties. You can see the customer value, which is the same as the configuration customer value. You can see the servers listed as managed servers only 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 37. Job Status on SA Jobs and Sessions**



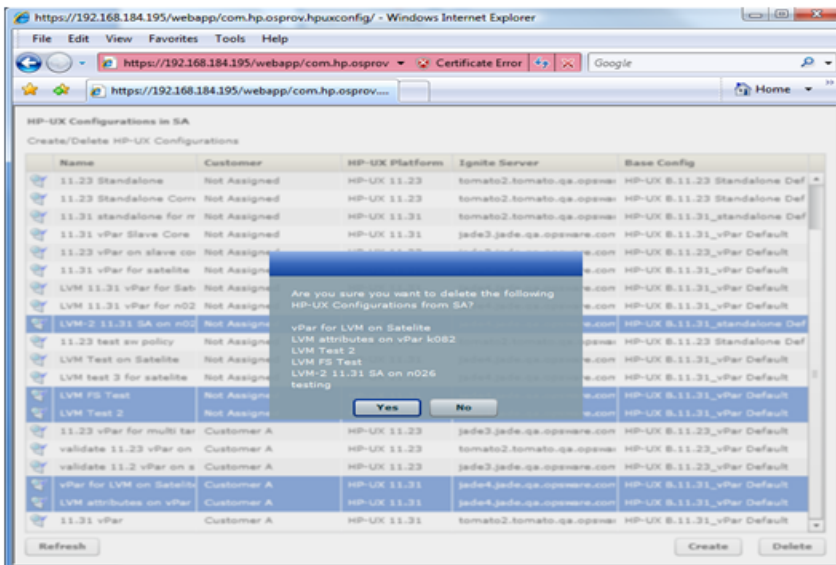
You can also find the specified Job Ticket ID in the notifications tab by double-clicking the job.

### 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 38. 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=>

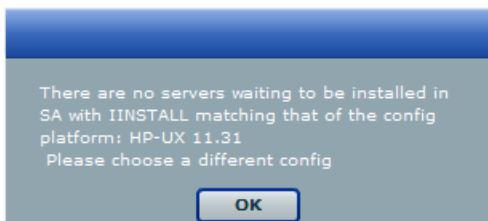
## Troubleshooting

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 39. No Servers Waiting to be Installed

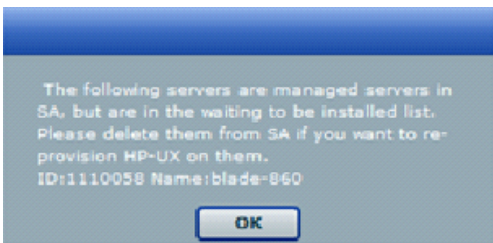


Ensure that you have the selected correct configuration.

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.

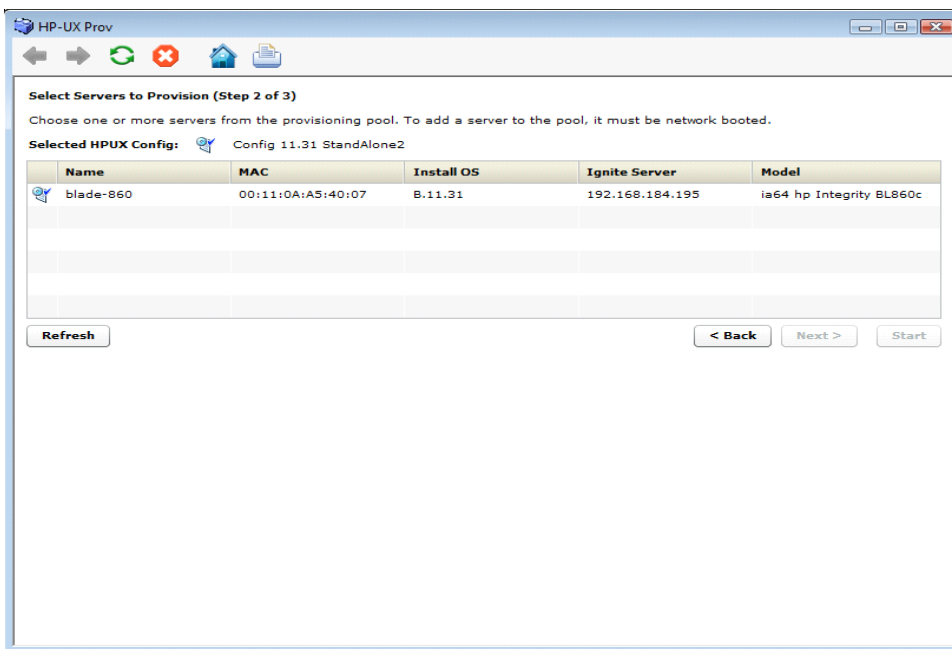
**Figure 40. Servers Waiting to be Installed are Managed Servers**



This warning message indicates that the listed servers are waiting for installation but are not candidates for re-provisioning because they are listed as Managed Servers in SA. To continue re-provisioning 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 User Guide: Server Automation*.

**Figure 41. List of Deleted Managed Servers for Re-provisioning**

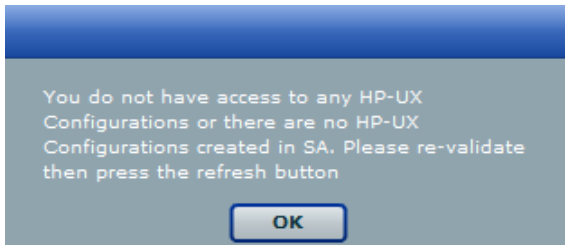


Once a server is deleted, it is not listed under the SA managed server list. Click **Refresh** in the HP-UX Provisioning APX window and the server should be listed under the unprovisioned server pool. Select the server and continue provisioning it.

**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 42. 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.

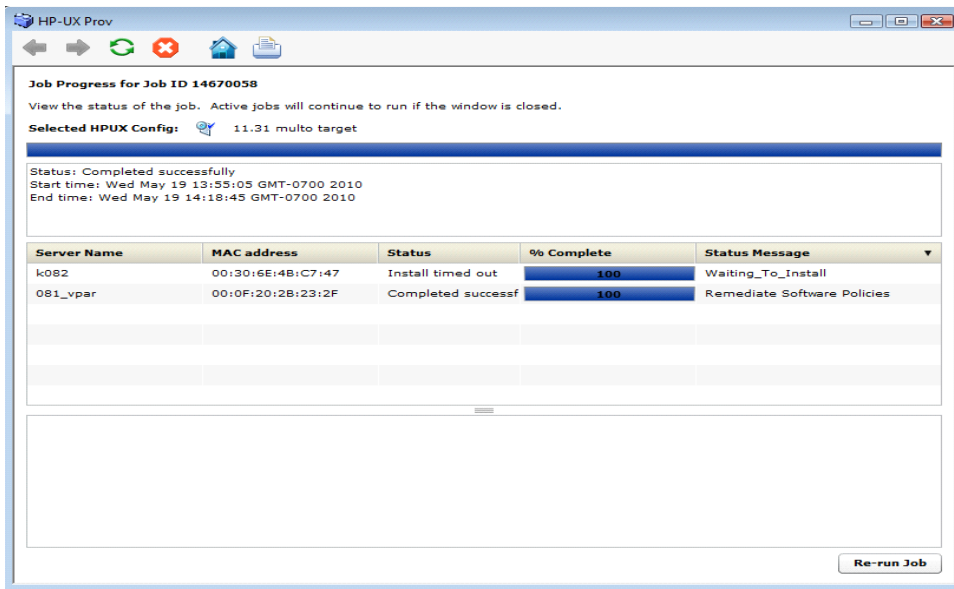
This can happen when you modify DHCP to provide a different hostname after having previously provisioned the client. Ignite UX reuses the directories in `/var/opt/ignite/clients/` it set up for a client (when it finds a client based on the MAC address), so the APX reuses that information. You can delete 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 43. Provisioning Not Initiated on Server**





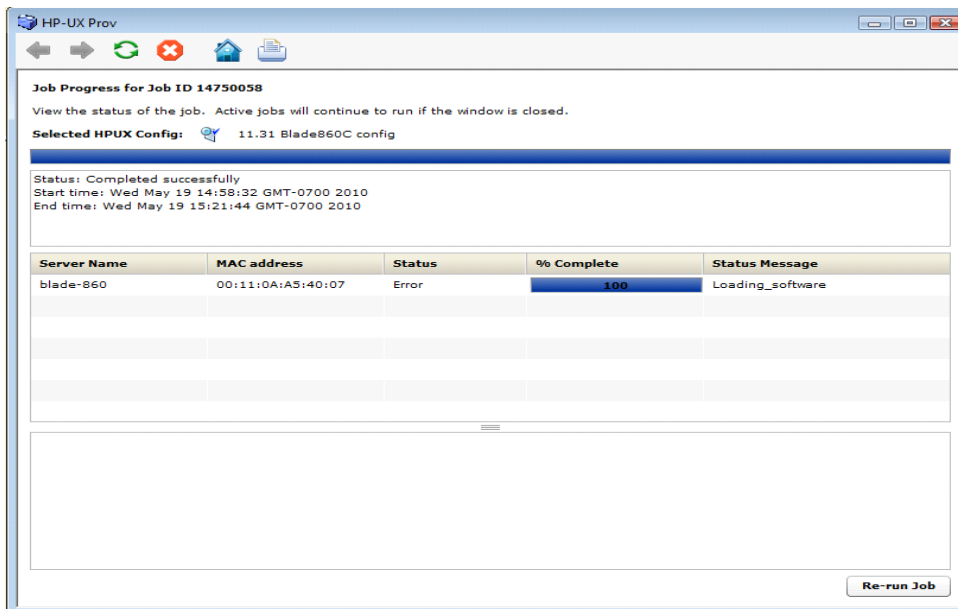
Ensure that the network connection and Ignite images are accessibility and run the APX again to initiate provisioning.

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 44. Golden Image Set Up Incorrectly**

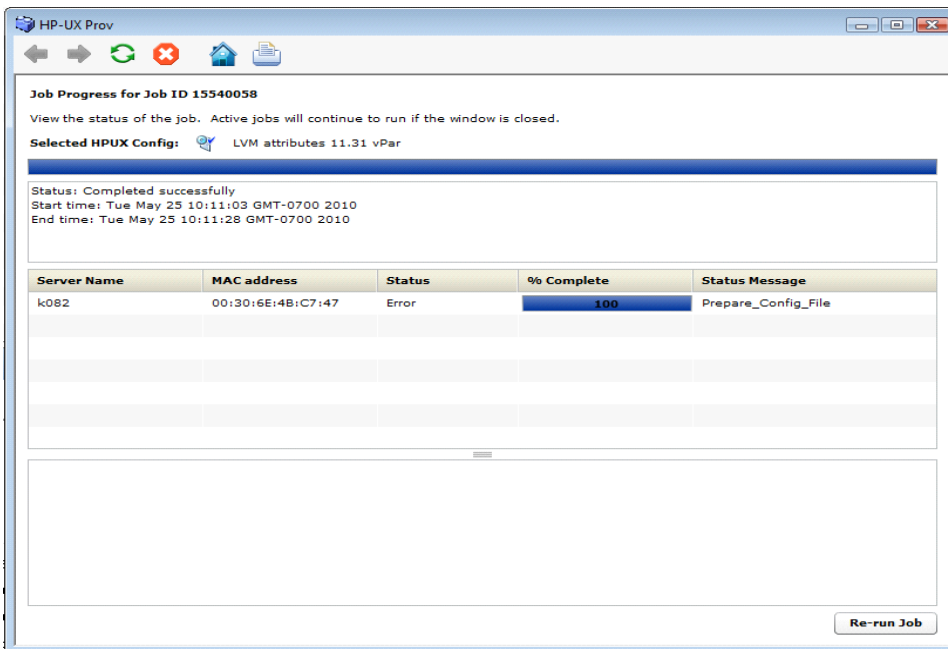


To resolve this, ensure that the Ignite-specific configuration file, Index file, and archives are correctly set up and pointing to correct locations. Also ensure that the network connection between the target and the Ignite server is accessible.

**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 45. 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.



# Index

---

## B

build customization scripts

Linux, overview 161

overview 155

requirements

for Linux 164

for Solaris 158

Solaris

overview 160

sample 160, 163

Build Manager

OS Build Agents, locating 119

## C

creating

OS sequence 178

custom attributes

Linux OS provisioning, setting for 172

Solaris OS provisioning, setting for 170

Windows OS provisioning, setting  
for 175

## D

deleting

media resource locators (MRLs) 133

DHCP

servers, booting with 117-118

Solaris servers, booting with 156

---

## E

examples

response file

for Windows NT 146

sample Solaris build customization  
script 160, 163

## F

firewall configuration 144

firewall configuration for OS  
Provisioning 144

Firewalls 144

## H

histories

viewing, changes in OS installation pro-  
files 154

## L

Linux

build customization scripts

overview 161

requirements for 164

setting, custom attributes for  
servers 172

## M

media resource locators (MRLs)

deleting 133

## O

operating systems

defining for OS provisioning 151

provisioning 16

OS Build Agents

Build Manager, locating 119

---

OS build process

default values for 170

OS installation profiles

histories, viewing 154

modifying packages in 154

overview 141

properties, changing 154

software, specifying 143

OS provisioning

Linux

custom attributes, setting up 172

OS installation profiles, preparing 151

Prepare Operating System Wizard 151

SA Client

creating an OS sequence 178

Solaris custom attributes, setting up 170

Windows custom attributes, setting  
up 175

OS Provisioning 144

OS sequence

attach device group 179

creating 178

set remediate policy 179

**P**

packages

modifying in OS installation profiles 154

Prepare Operating System Wizard 151

properties, OS installation profiles, changing  
for 154

**R**

Red Hat Linux 144

response files

example

for Windows NT 146

**S**

scripts

Linux build customization scripts 164

Linux servers, customizing build 161

Solaris build customization scripts, require-  
ments for 158

Solaris servers, customizing build 160

setup for servers

operating systems for provisioning 151

software

specifying in OS installation profiles 143

Solaris

build customization scripts

overview 160

sample 160, 163

custom attributes, setting for Solaris serv-  
ers 170

requirements for build customization  
scripts 158

**V**

viewing

changes for OS installation profiles 154

**W**

Windows servers

sample response file

for Windows NT 146

setting custom attributes for 175

wizards

Prepare Operating System 151

# Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on User Guide: Provisioning (Server Automation 10.23)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [hpe\\_sa\\_docs@hpe.com](mailto:hpe_sa_docs@hpe.com).

We appreciate your feedback!