## Insight Control server provisioning

### All About OS Build Plans and Steps

**Table of contents**

<table>
<thead>
<tr>
<th>Summary</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OS Build Plans</strong></td>
<td>7</td>
</tr>
<tr>
<td>ProLiant Combination</td>
<td>7</td>
</tr>
<tr>
<td>ProLiant Hardware</td>
<td>7</td>
</tr>
<tr>
<td>ProLiant Operating System</td>
<td>7</td>
</tr>
<tr>
<td>ProLiant Software</td>
<td>7</td>
</tr>
<tr>
<td><strong>OS Build Plan Steps</strong></td>
<td>8</td>
</tr>
<tr>
<td>Run Script</td>
<td>8</td>
</tr>
<tr>
<td>OGFS</td>
<td>8</td>
</tr>
<tr>
<td>Python</td>
<td>8</td>
</tr>
<tr>
<td>Unix</td>
<td>8</td>
</tr>
<tr>
<td>Windows BAT</td>
<td>8</td>
</tr>
<tr>
<td>Windows VBScript</td>
<td>8</td>
</tr>
<tr>
<td>Powershell</td>
<td>8</td>
</tr>
<tr>
<td>Deploy Package</td>
<td>8</td>
</tr>
<tr>
<td>Deploy Configuration File</td>
<td>8</td>
</tr>
<tr>
<td>Capture Configuration File</td>
<td>8</td>
</tr>
<tr>
<td><strong>Working with OS Build Plans</strong></td>
<td>9</td>
</tr>
<tr>
<td>Modifying Configuration Files</td>
<td>9</td>
</tr>
<tr>
<td>Adding Scripts</td>
<td>9</td>
</tr>
<tr>
<td>Modifying Build Plan Step Parameters</td>
<td>9</td>
</tr>
<tr>
<td>Combining Build Plans</td>
<td>10</td>
</tr>
<tr>
<td><strong>Detailed Breakdown of Sample Build Plans</strong></td>
<td>10</td>
</tr>
<tr>
<td>Sample Build Plan: Windows Scripted Install</td>
<td>10</td>
</tr>
<tr>
<td>Sample Build Plan: Windows Image Capture</td>
<td>13</td>
</tr>
<tr>
<td>Sample Build Plan: Windows Image Install</td>
<td>15</td>
</tr>
<tr>
<td>Sample Build Plan: Linux Scripted Install</td>
<td>17</td>
</tr>
<tr>
<td>Sample Build Plan: ESXi Scripted Install</td>
<td>20</td>
</tr>
<tr>
<td>Sample Build Plan: ProLiant Hardware Configuration</td>
<td>22</td>
</tr>
<tr>
<td>Sample Build Plan: ProLiant Software Install</td>
<td>23</td>
</tr>
<tr>
<td>Sample Build Plan: Combination Hardware, Windows Scripted Install, and SPP</td>
<td>25</td>
</tr>
<tr>
<td>How these build plans were combined</td>
<td>26</td>
</tr>
<tr>
<td>What this build plan does</td>
<td>26</td>
</tr>
</tbody>
</table>
Install Windows SPP 44
Integrate HP SA Agent 45
Integrate Linux HP SA Agent 45
Manage iLO Configuration 45
Manage Smart Array Configuration 46
Manage System Configuration 46
Monitor Installation 46
Prepare Disks on HP ProLiant Gen8 47
Prepare Windows for Image Capture 47
Prevent WIM File Overwrite 47
Reboot 47
Reboot to Apply BIOS Changes and Power Off 48
Remap Windows Drives 48
Reset System Configuration 48
Run Windows 2008 R2 x64 Setup 49
Run Windows 2008 x64 Setup 49
Run Windows 2012 x64 Setup 49
Set Media Source 49
Set One Time PXE Boot 50
Uninstall HP Agents and Utilities 50
Unmap Network Drive 50
Unmount All Boot Disk Partitions 50
Unmount Intelligent Provisioning WinPE Volume 50
Update Firmware Using SPP 51
Validate Custom Attributes 51
Validate Gateway Setting for Static Network Configuration 51
Validate ImageX Package Contents 52
Wait for ESXi installation 52
Wait for HP SA Agent 52
Configuration Files 53
Summary 53
ESXi 5.0 U1 Kickstart 53
ESXi 5.0 U2 Kickstart 53
ESXi 5.1 Kickstart 53
ESXi 5.1 U1 Kickstart 53
ESXi 5.5 Kickstart 53
iLO Configuration - Set Minimum Password Length 53
ImageX Configuration - Exclude Boot Directory 53
RHEL 5.9 x64 en_us Kickstart 54
RHEL 6.3 x64 en_us Kickstart 54
RHEL 6.4 x64 en_us Kickstart 54
RHEL 6.4 x64 KVM en_us Kickstart 54
SLES 11 SP2 x64 en_us Autoyast 54
SLES 11 SP3 x64 en_us Autoyast 54
Smart Array Configuration - Delete RAID Logical Volumes 54
Smart Array Configuration - Set RAID 1 54
Smart Array Configuration - Set RAID 5 54
System ROM Configuration - Enable Boot from SAN 54
System ROM Configuration – Enable Virtualization 54
Windows 2008 SP2 DataCenter x64 en_us Unattend 54
Windows 2008 SP2 Enterprise x64 en_us Unattend 54
Windows 2008 SP2 Standard x64 en_us Unattend 54
Windows 2008 SP2 Web Server x64 en_us Unattend 54
Windows 2008 R2 SP1 DataCenter x64 en_us Unattend 55
Windows 2008 R2 SP1 Enterprise x64 en_us Unattend 55
Windows 2008 R2 SP1 Standard x64 en_us Unattend 55
Windows 2008 R2 SP1 Web Server x64 en_us Unattend 55
Windows 2012 DataCenter x64 en_us Unattend 55
Windows 2012 Standard x64 en_us Unattend 55

Packages 56
Summary 56
ESXi Installation Utilities 56
GRub Boot Loader x86 56
Hponcfg for Windows x64 with One Time PXE Boot 56
ImageX 56
LinuxPE add-on packages 56
LinuxPE HBA add-on packages 56
ProLiant Drivers for RHEL 5.9 x64 - yyyy.mm 57
ProLiant Drivers for RHEL6.3 x64 - yyyy.mm 57
ProLiant Drivers for RHEL6.4 x64 - yyyy.mm 57
ProLiant Drivers for SLES 11 SP2 x64 - yyyy.mm 57
ProLiant Drivers for SLES 11 SP3 x64 - yyyy.mm 57
ProLiant Drivers for Windows 2008 x64 - yyyy.mm 57
ProLiant Drivers for Windows 2008 R2 x64 - yyyy.mm 57
ProLiant Drivers for Windows 2012 - yyyy.mm 57
ProLiant Scripting Toolkit for Linux x64 57

Appendix A 57
New Changes for IC server provisioning Release 7.2.2 57
Custom Attribute added for several ProLiant HW build plans 57
Script step added to ESXi build plans for network gateway validation 57
New ProLiant Driver package naming to include SPP version 57
Windows build plans contain comment information regarding disk partitioning in the unattend answer files 57
Linux build plans have firewall disabled in default kickstart or autoyast answer files 58
Modifications to the Erase Server Disk build plan 58

For more information 58
Insight Control (IC) server provisioning provides OS Build Plans, scripts, packages, and configuration files that are used to deploy operating systems, configure hardware, and update firmware.

OS build plans are the way tasks get done in IC server provisioning. These are the items you actually run to cause actions like installing a server or updating firmware to happen. OS build plans are simply a collection of ordered steps and parameters associated with those steps that when placed together, in the proper order, can perform just about any action you require. IC server provisioning comes ready to run, with sample build plans and build plan steps that are designed to work right out of the box. These sample build plans are very important, because they demonstrate the steps needed to perform the most common deployment related operations. Although you can create your own build plans from scratch, it is expected that most users will start with one of the provided samples and modify it to perform the functions they need.

This document provides detailed information on
- the architecture of OS Build Plans,
- what OS Build Plans, scripts, packages, and configuration files are available in the appliance library,
- how to use the scripts parameters, and
- best practices when using these objects.

This technical white paper presumes that the reader is familiar with IC server provisioning. For IC server provisioning installation information, refer to HP Insight Control Server Provisioning Installation Guide. IC server provisioning documentation can be found at www.hp.com/go/insightcontrol/docs.

OS Build Plans

Although build plans are referred to as OS Build Plans, they do much more than operating system deployment. For example, build plans can also be used to
- Configure a target server’s hardware.
- Capture a target server’s hardware configuration, so that the same configuration can later be applied to other servers.
- Update the firmware on a target server.
- Install software on a target server with a running operating system.

IC server provisioning provides four types of sample build plans:

**ProLiant Combination**
Beginning with IC server provisioning release 7.2.2, build plans labeled with ProLiant COMBO perform a combination of functions on target servers, such as hardware-related configurations, deploying an operating system, and installing software.

**ProLiant Hardware**
Build plans labeled with ProLiant HW perform hardware-related functions on target servers such as booting the target server to the proper service OS, or capturing or configuring hardware settings.

**ProLiant Operating System**
Build plans labeled with ProLiant OS deploy an operating system to target servers either via scripted or image installation.

**ProLiant Software**
Build plans labeled with ProLiant SW perform functions on target servers to update the firmware or install/update software on target servers running a production operating system.
OS Build Plan Steps

Build plans are made up of a series of build plan steps that execute in order to perform the actions. Four types of steps are available: Run Script, Deploy Package, Deploy Configuration File, and Capture Configuration File.

Run Script

The run script step is the key component of the product, and represents the vast majority of steps used in build plans. This step type causes a script to be executed, either on the target server or on the appliance. IC server provisioning comes with an extensive library of scripts that perform many of the most common tasks you will need when creating build plans. In addition, you can create your own scripts based on the ones HP provides or entirely from scratch.

IC server provisioning supports the following script types:

**OGFS**
OGFS scripts control the HP Server Automation engine that is inside the IC server provisioning appliance. These are the only scripts that execute on the appliance. All other script types run on the target server. Most of the OGFS scripts shipped with the appliance come from Server Automation, are written in Python, and are not meant to be modified in any way. They provide vital functions like booting target servers, monitoring tasks, and manipulating data. HP does not recommend creating OGFS scripts unless you have advanced knowledge of Server Automation.

**Python**
Python scripts execute on the target server. This is the only script type that can be run on either Windows or Linux systems.

**Unix**
These are standard Unix/Linux shell scripts that execute on the target server.

**Windows BAT**
These are standard Windows batch scripts that execute on the target server.

**Windows VBScript**
These are standard Windows Visual Basic scripts that execute on the target server.

**Powershell**
IC server provisioning does not currently support PowerShell as a script type. However, a Windows batch script can be created

- that dynamically generates the PowerShell script and then runs the script on the target server,
- pipes the PowerShell content into the PowerShell interpreter to run it,
- or copies a PowerShell script from the Media Server to the target server and then runs it.

Deploy Package

In IC server provisioning, packages are zip files stored on the appliance. When the deploy package step is used, the zip file is transferred to the target server and uncompressed into the specified location. Pre-installation and post-installation scripts may also be specified with the package that will execute before or after the package is saved to the target server and the files extracted. All the packages on your appliance are provided by HP. They are typically things like driver bundles and software libraries needed for installing on ProLiant servers. At this time, there is no option to allow you to upload your own packages or save and modify copies on the appliance, although that functionality is planned for a future IC server provisioning release.

Deploy Configuration File

Configuration files are text files stored on the appliance that 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. These 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 or create your own.

Capture Configuration File

This step allows you to capture a text file from the target server and upload it to your appliance database so that it can later be used as part of a deploy configuration file step. This capture step is typically used to capture the configuration
of a hardware component, so that the same configuration can be applied to other servers. Use caution with this step as you can easily create a large number of configuration files if you run a build plan against many servers.

**Working with OS Build Plans**

All of the HP-provided build plans and build plan steps are read only which gives you a consistent and reliable source for working samples. Although most of the HP-provided build plans will work without modification, it is highly unlikely the HP-provided build plans will exactly meet your requirements. It is expected that you will create your own build plans based on the examples we have provided.

Since the HP-provided build plans and steps are read only, here is an example of how to modify a build plan to use a new configuration file:

1. Open the HP-provided configuration file to be modified and make a copy using *Actions > Save as*.
2. Modify the new configuration file.
3. Open the HP-provided build plan to be modified with the new configuration file and make a copy using *Actions > Save as*.
4. Edit the newly copied build plan.
   a. In the Steps section of the edit page, find the HP sample configuration file to be replaced and select the *Edit* icon for that step. This will open the edit screen for that step.
   b. If there are parameters associated with this step, select them and copy them to the paste buffer, since they will clear when the new step is selected.
   c. Click the drop down and select the newly modified configuration file created in Step 2.
   d. If there were parameters with this build plan step, paste them back into the text box.
   e. Click *OK* to save the step change.
   f. Click *OK* to save the new build plan.

It is important to understand that the steps listed in a build plan are references or links. That means that if the contents of a script or a configuration file are modified, every build plan that uses that step will change with that modification. However, the parameters that are specified for that build plan step are stored with the build plan and can be changed without affecting the other build plans.

Modifying a build plan should not require a lot of changes. Most of the steps in the sample build plans are required, because they do actions such as boot the server, setup the installation and install agents. The following sections cover some of the most common types of changes that you may need to make. The detailed build plan descriptions in later sections will give indications as to which steps commonly need to be changed.

**Modifying Configuration Files**

The default build plans come with sample configuration files. These might be sample unattended files for installing Windows, kickstart or autoyast files for installing Linux, or hardware configuration files for configuring a system option. Additionally, sample configuration files are provided for each supported Windows edition. By far, the most common modification will be to replace the HP sample configuration file with a customized configuration file.

**Adding Scripts**

Another common modification is to perform additional tasks on the target server after it has been installed. You can do this by creating scripts, and then adding those scripts to the end of the build plan after the operating system installation is complete.

**Modifying Build Plan Step Parameters**

Some build plan steps are controlled by the parameters associated with that step. You may wish to change these parameters to better suit your needs. For most of the HP-provided scripts, a summary of parameters for that script are listed in the script’s description field.
Combining Build Plans

Combining multiple build plans into one build plan may be desired if a series of build plans are frequently run in the same order. For example, to configure the system ROM, setup the RAID controller, and then install the operating system. In most cases, combining build plans is as simple as combining all of the steps and parameters from the multiple build plans and putting them into one large build plan in the same order. Once all of the steps are put together, you may find it beneficial to make slight adjustments to the combined build plan that will make it more efficient or provide better error checking. HP provides some sample build plans that can be used as an example of combining build plans and the types of modifications one might make to them once they are combined. See the detailed breakdown of the combo build plans in the following section for descriptions of such modifications.

Detailed Breakdown of Sample Build Plans

Looking at a build plan for the first time can be somewhat intimidating. Some have many steps you may not fully understand, which can make modifying and troubleshooting them very challenging.

This section provides information on the various steps of the HP-provided build plans. It will explain what the steps do, the order they are in, if they are required, and if they are meant to be modified. It will also explain the general methods used to perform the build plan’s functionality. It is suggested to review all the build plan samples in this section since information that appears in multiple build plans will not be explained more than once.

NOTE: These examples are just a sampling of all the build plans HP provides and may not exactly match the build plans on the appliance, but the information provided will still be valid.

Sample Build Plan: Windows Scripted Install

Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Parameters</th>
<th>Step Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Validate Custom Attributes</td>
<td>--custAttrNames &quot;ProductKey_Win2012-Std-x64&quot;</td>
<td>OGFS</td>
</tr>
<tr>
<td>2</td>
<td>Check iLO Service</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>3</td>
<td>Boot</td>
<td>--serviceOS=winpe64</td>
<td>OGFS</td>
</tr>
<tr>
<td>4</td>
<td>Decommission Server</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>5</td>
<td>Wait for HP SA Agent</td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td>OGFS</td>
</tr>
<tr>
<td>6</td>
<td>Set Media Source</td>
<td>@__OPSW-Media-WinUser@@__OPSW-Media-WinPassword@@__OPSW-Media-WinPath#@z</td>
<td>Python</td>
</tr>
<tr>
<td>7</td>
<td>Prepare Disks on HP ProLiant Gen8</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>8</td>
<td>Create Windows System Drive</td>
<td></td>
<td>Windows .BAT</td>
</tr>
<tr>
<td>9</td>
<td>Windows 2012 Standard x64 en_us Unattend</td>
<td>x:\Windows\Temp\Unattend.xml</td>
<td>Deploy configuration file</td>
</tr>
</tbody>
</table>
Steps 1 and 2 – Early error detection
These first two steps help catch errors that might affect the running of the build plan later on. Most of the HP-provided build plans contain a varying number of steps like this example. The idea is that it’s better to catch an error right away than after the build plan has been running for a while.

- **Validate Custom Attributes** – This step checks for the existence of custom attributes (CAs) that are required for the build plan to run, and it will verify that the custom attribute has a value. It will check for any CAs specified in the parameters section. It does not validate the custom attribute value. For this build plan, it is checking to ensure that the product key for this particular OS has been defined. If this build plan step was not included and the product key custom attribute was not defined, the build plan would fail toward the end of the installation. This step catches that potential mistake right away so that it can be corrected more quickly. If it is preferred not to do this type checking up front, this step can safely be removed from any build plan. Also note that if the build plan is modified to install a different edition of the OS, this step will require modification to include the matching custom attribute for that version.

- **Check iLO Service** – This step verifies that the server on which the build plan is running on has an iLO management processor associated with it in the appliance database. This is important, because the boot step later in the build plan needs to communicate with the iLO to boot the server. If the target server was discovered by PXE booting, the iLO registration for that server happens after the server appears in the appliance. The check iLO service step will wait up to 15 minutes for that registration to complete.

Steps 3 to 5 – Boot the server for provisioning
The next set of steps are used to boot the server into the required service OS and test the status of the server such that it can be provisioned. Once these steps are done, the server is in maintenance, and ready to start the provisioning process.

- **Boot** – The boot step is used to get the server into the appropriate service OS for provisioning. The desired service OS is specified in the parameters of the step, in this case, winpe64. When this step runs, it first checks to see if the server is already in that service OS, and if it is, the step exits. If the server is not in the right service OS, it contacts the iLO to power down the server, set the required boot parameters, and power the server back on. Note that once the boot is initiated, the step exits. The **Wait for HP SA Agent** step below will wait for the boot to complete.

- **Decommission Server** – This step only has an effect if a server is being reprovisioned that was already installed and running the HP SA agent. It tells the appliance that this target server is no longer going to be used as a production server and that it can be booted to maintenance and reprovisioned. If the target server was not previously installed and managed, this step does nothing. If this step is removed from the build plan plan, and you choose to reprovision a server, you will need to run either one of the **Prepare Server for Reprovisioning** build plans, or delete the target server from the appliance and re-add it. Note that this step must always come between the **Boot** and **Wait for HP SA Agent** steps.
• **Wait for HP SA Agent** – A boot step will always have this step after it. It waits for the boot to complete and the agent to contact the appliance. It then verifies that the server is in the appropriate mode as specified by the parameter, in this case “maintenance”. The other parameters specify the minimum and maximum wait times for this step. In this case, the step will wait 3 minutes, and then start checking if the agent is running on the target server. If communication with the agent has not been established after 20 minutes, the build plan fails. Depending on the circumstances, it may be necessary to adjust these parameters.

**Steps 6 to 12 – Stage the installation**

These steps perform all the work of setting up for the installation by partitioning the system drive, copying files, putting drivers in place, and preparing the unattended file.

• **Set Media Source** – This is the step that points the build plan to the OS distribution media on the media server. The parameter for this step is made up of three special custom attributes all strung together. These are hidden custom attributes that correspond to the information specified in the Media Server Settings page. These custom attributes should always be used together like this when installing a Windows OS. At run time, these custom attributes are substituted, and the result is a URI that takes this form:

```
smb://username:password@media-server-IP/share-name#drive-letter
```

where `username:password` are the credentials for the media server, and `drive-letter` is the drive letter the share will be mounted. Optionally, all three custom attributes can be replaced with customized values as long as it conforms to URI specification above.

For Windows build plans, the **Set Media Source** step actually does the mapping of the network drive into the service OS.

• **Prepare Disks on HP ProLiant Gen8** – HP ProLiant Gen8 and newer servers have multiple embedded flash drives that are part of the new advanced features of these platforms. Some OS installation programs can see these flash drives and incorrectly try to install the operating system to them causing the installation to fail. This step does some work to help the OS installation program identify these drives and not install to them. The SystemDiskNumber custom attribute is modified by this step if not already set. For HP ProLiant servers prior to Gen8, this step has no effect.

• **Create Windows System Drive** – This step uses diskpart to create a small temporary partition on the hard drive to store installation files.

• **Windows 2012 Standard x64 en_us Unattend** – This is a deploy configuration file step. It writes the HP-provided unattend file to the target server’s ram drive. It is recommended to replace this step with a customized unattend file using the HP-provided configuration files as a template since these template files contain the Custom Attribute syntax. The path specified by the parameter is where the file is written. Subsequent build plan steps expect to find this file in that location, so it is recommended that the path not be changed. **NOTE:** The partitioning of the boot disk is performed in the previous **Create Windows System Drive OSBP** step. Refer to the **Create Windows System Drive** script description in the Scripts section within this document for important information about disk partitioning.

• **Inject Required Unattend.xml Settings** – This is a script that adds required items to the unattend file to make it work properly with the appliance. This step is required and always comes right after the step that writes out the unattend file.

• **Inject Personalization Settings** – This step is used when static IP addressing information is specified as part of the installation in the Run OS Build Plan page. Setting network information there causes a server-level custom attribute named, hpsa_netconfig, to be created and assigned to each server being provisioned. The **Inject Personalization Settings** step checks for the existence of that custom attribute. If it exists, the network information is read and the unattend file is modified to include the static IP information. This step should always follow the **Inject Required Settings** step. **Note:** The hpsa_netconfig custom attribute is not removed after it is set, so it could be there if another build plan is run against the same target server.

• **ProLiant Drivers for Windows 2012** – This is a zip file containing the latest ProLiant drivers for this operating system. They are placed in the directory specified by the parameter, which is where the Windows OS installer is expecting to find them.

**Steps 13 to 16 – Perform the installation and post-install tasks**

These final steps actually perform the OS installation, add the production agent, and perform the final reboot of the server.
• **Run Windows 2012 x64 Setup** – This is the step that runs the Windows installation. It is a batch script that calls the Windows setup.exe program to perform the installation. The parameter for this step is the full path to the setup.exe program on the Media Server. Note that it references drive Z, which is the drive specified in the Set Media Source step. If the drive letter in Set Media Source is changed, it needs to be changed here as well.

• **Integrate HP SA Agent** – This step adds the HP Server Automation agent to the newly installed operating system.

• **Reboot** – Initiates a reboot of the server to finalize the installation. Note that the Reboot step is used here instead of the Boot step. The Reboot step is specifically meant for rebooting a server to its local disk and back into production.

• **Wait for HP SA Agent** – This is the same step as before, but this time it is waiting for the agent to register in production mode. Also, note that the atMost parameter is now 30 minutes since booting Windows for the first time after an installation takes extra time while automatic configuration is performed.

Sample Build Plan: Windows Image Capture

Table 2 – ProLiant OS – Windows 2012 Standard x64 Image Capture build plan sample

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Parameters</th>
<th>Step Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Validate Custom Attributes</td>
<td>--custAttrNames &quot;WimFileName&quot;</td>
<td>OGFS</td>
</tr>
<tr>
<td>2</td>
<td>Check iLO Service</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>3</td>
<td>Unmap Network Drive</td>
<td>--driveLetter=Z</td>
<td>Python</td>
</tr>
<tr>
<td>4</td>
<td>Set Media Source</td>
<td>@__OPSW-Media-WinUser@@__OPSW-Media-WinPassword@@__OPSW-Media-WinPath@#z</td>
<td>Python</td>
</tr>
<tr>
<td>5</td>
<td>Prevent WIM File Overwrite</td>
<td>&quot;Z:\Images@WimFileName@&quot;</td>
<td>Windows .BAT</td>
</tr>
<tr>
<td>6</td>
<td>ImageX</td>
<td>%SystemDrive%\HPPROVTEMP</td>
<td>Deploy package</td>
</tr>
<tr>
<td>7</td>
<td>Validate ImageX Package Contents</td>
<td></td>
<td>Windows .BAT</td>
</tr>
<tr>
<td>8</td>
<td>Uninstall HP Agents and Utilities</td>
<td></td>
<td>Windows VBScript</td>
</tr>
<tr>
<td>9</td>
<td>Prepare Windows for Image Capture</td>
<td></td>
<td>Windows .BAT</td>
</tr>
<tr>
<td>10</td>
<td>Boot</td>
<td>--serviceOS=winpe64</td>
<td>OGFS</td>
</tr>
<tr>
<td>11</td>
<td>Wait for HP SA Agent</td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td>OGFS</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td>Language</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Prepare Disks on HP ProLiant Gen8 OGFS</td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Remap Windows Drives Python</td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>ImageX Deploy package</td>
<td>Windows .BAT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X:\Windows\System32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Validate ImageX Package Contents</td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Windows .BAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Unmap Network Drive Python</td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--driveLetter=Z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Set Media Source Python</td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td></td>
<td>@__OPSW-Media-WinUser@@__OPSW-Media-WinPassword@@__OPSW-Media-WinPath#@z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Capture Windows Image Windows .BAT</td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Z:\Images@WimFileName@” @CaptureDrive:C:@</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Important:** By default, the ImageX zip package used in build plan steps 6 and 14 does not contain the imagex.exe utility. Using the IC server provisioning WinPE Image Generation utility, which is available for download from the Settings > DHCP page, a zip package containing WinPE, as well as the required imagex.exe utility is created. When this zip package is uploaded to the appliance in the Settings > DHCP page, using the WinPE Upload button, the imagex.exe utility will be automatically inserted into the ImageX package.

**Steps 1 and 2 – Early error detection**
These first two steps help catch errors that might affect the running of the build plan later on. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed descriptions in the Steps 1 and 2 – Early error detection section for that table.

**Steps 3 to 5 – Test for an existing image**
These steps provide more early error detection. They ensure that a previously captured WIM image file is not accidentally overwritten. If you want this capture build plan to be able to overwrite a previous image, all three of these steps can be removed from the build plan.

- **Unmap Network Drive** – This step unmaps the network drive to ensure that a drive used by a previous build plan is available for use by the current build plan.
- **Set Media Source** – This step mounts the file share from the Media Server so that the following, Prevent WIM File Overwrite step can check if there is already an image on the Media Server with the same file name specified in the @WimFileName@ custom attribute.
- **Prevent WIM File Overwrite** – This step ensures that a WIM file that was previously captured is not accidentally overwritten. The @WimFileName@ custom attribute used by the build plan contains the name of the file where the captured image will be saved. If the intent is truly to replace a previously captured image using the same file name, remove the file for the old image from the Media Server.

**Steps 6 and 7 – Test for valid ImageX package**
These steps ensure that the ImageX package contains the imagex.exe utility. Like the early error detection build plan steps, these steps are for early detection that the imagex.exe utility was uploaded properly to the appliance and fails the build plan before it is rebooted into the service OS where the imagex.exe utility will actually be run.

- **ImageX** – This is a zip package containing the imagex.exe utility that is needed to capture the Windows OS.
- **Validate ImageX Package Contents** – Verifies that the ImageX package contains the required imagex.exe utility.

**Steps 8 and 9 – Prepare production OS for capture**
These steps ensure that the production OS is left in a state where it can be captured and will not cause any conflicts when it is deployed to a different target server.
• **Uninstall HP Agents and Utilities** – Uninstalls HP Agents and Utilities stores server-specific information and, therefore, must not be included in the captured image. These agents or utilities will not operate properly when deployed to a different target server.

• **Prepare Windows for Image Capture** – This step removes unique information from the Windows OS using sysprep so that the image can be safely used on a different target server, and configures the Windows installation to boot to Out-of-box Experience (OOBE) the next time that the server boots.

**Steps 10 and 11 – Boot the server for capture**
The next set of steps is used to boot the target server into WinPE for the purpose of capturing the image.

• **Boot** – Reboots the target server into WinPE.

• **Wait for HP SA Agent** – Waits for WinPE to boot and the SA agent to be available.

**Steps 12 and 13 – Adjust the drives**
These steps make any necessary adjustments to the drive letters due to the presence of embedded flash drives on HP ProLiant Gen8 servers.

• **Prepare Disks on HP ProLiant Gen8** – Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the **Steps 6 to 12 – Stage the installation** section.

• **Remap Windows Drives** – For cases where the previous **Prepare Disks on HP ProLiant Gen8** step hid the embedded flash drives, this step will remap the drive letters, so that the boot disk is assigned the “C:” drive, and the remaining disks are assigned to sequential drive letters.

**Steps 14 and 15 – Prepare for image capture**
These steps ensure that the ImageX package contains the imagex.exe utility and lands the files in the service OS to be used for the capture. Refer to Table 2 – ProLiant OS – Windows 2012 Standard x64 Image Capture build plan sample and the detailed description in the **Steps 6 and 7 – Test for valid ImageX package** section.

**Steps 16 to 18 – Create WIM Image on the Media Server**
These steps mount the file share from the Media Server and capture the WIM image onto the Media Server. The target server is left in maintenance, because the Prepare Windows for Image Capture step left the server in an OOBE state and, therefore, cannot be booted to production.

• **Unmap Network Drive** – This step unmaps the network drive to ensure that a drive used by a previous build plan is available for use by the current build plan.

• **Set Media Source** – This step mounts the file share from the Media Server so that the captured image can be saved onto the Media Server.

• **Capture Windows Image** – Creates a WIM image of the captured Windows OS, which it stores on the Media Server in a file whose name is specified by the WimFileName custom attribute.

**Sample Build Plan: Windows Image Install**

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Parameters</th>
<th>Step Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Validate Custom Attributes</td>
<td>--custAttrNames &quot;WimFileName ProductKey_Win2012-Std-x64&quot;</td>
<td>OGFS</td>
</tr>
<tr>
<td>2</td>
<td>Check iLO Service</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>3</td>
<td>Boot</td>
<td>--serviceOS=winpe64</td>
<td>OGFS</td>
</tr>
<tr>
<td>4</td>
<td>Decommission Server</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>Step</td>
<td>Task Description</td>
<td>Script/Command</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Wait for HP SA Agent</td>
<td><code>--maintenance --atLeast=3 --atMost=20</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Prepare Disks on HP ProLiant Gen8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Create Windows System Drive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Set Media Source</td>
<td>Python @__OPSW-Media-WinUser@__OPSW-Media-WinPassword@__OPSW-Media-WinPath@#z</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ImageX Deploy package</td>
<td>X:\Windows\System32</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Validate ImageX Package Contents</td>
<td>Windows .BAT</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Apply WIM Image</td>
<td>Windows .BAT &quot;Z:\images@WimFileName@&quot;</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Windows 2012 Standard x64 en_us Unattend Deploy configuration file</td>
<td>@SystemDrive:C@:\Windows\Panther\unattend.xml</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Inject Required Unattend.xml Settings</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Inject Personalization Settings</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Integrate HP SA Agent</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Reboot</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Wait for HP SA Agent</td>
<td>OGFS <code>--production --atLeast=3 --atMost=30</code></td>
<td></td>
</tr>
</tbody>
</table>

**Steps 1 and 2 – Early error detection**
These first two steps help catch errors that might affect the running of the build plan later on. Refer to Table 2 – ProLiant OS – Windows 2012 Standard x64 Image Capture build plan sample and the detailed description in the *Steps 6 and 7 – Test for valid ImageX package* section.

**Steps 3 to 5 – Boot the server for provisioning**
The next set of steps are used to boot the server into the required service OS and rest the status of the server such that it can be provisioned. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed descriptions in the *Steps 3 to 5 – Boot the server for provisioning* section for that table.

**Steps 6 to 8 – Stage the installation**
These steps perform all the work of setting up for the installation by partitioning the system drive, mapping a drive to the media server’s file share, and copying the imagex.exe utility onto WinPE. Refer to Table 2 – ProLiant OS – Windows 2012 Standard x64 Image Capture build plan sample and the detailed description in the *Steps 6 and 7 – Test for valid ImageX package* section.
Steps 9 and 10 – Prepare for image deployment
These steps ensure that the ImageX package contains the imagex.exe utility and lands the files in the service OS to be used for the capture. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the Steps 6 to 12 – Stage the installation section.

Steps 11 to 15 – Install and Customize Image
These steps install the image and perform any customization.

- **Apply WIM Image** – This step takes a previously captured image that is saved on the Media Server and installs it on the target server's boot disk.
- **Windows 2012 Standard x64 en_us Unattend** – Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the Steps 6 to 12 – Stage the installation section. The configuration file used with the image installation build plan is the same as with the scripted install.
- **Inject Required Unattend.xml Settings** – Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the Steps 6 to 12 – Stage the installation section.
- **Inject Personalization Settings** – Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the Steps 6 to 12 – Stage the installation section.
- **Integrate HP SA Agent** – This step adds the HP Server Automation agent to the newly installed operating system.

Steps 16 and 17 – Boot the server to production
These final steps boot the server into production.

- **Reboot** – Initiates a reboot of the server to finalize the installation. Note that the Reboot step is used here instead of the Boot step. The Reboot step is specifically meant for rebooting a server to its local disk and back into production.
- **Wait for HP SA Agent** – This is the same step as before, but this time it is waiting for the agent to register in production mode as opposed to maintenance. Also, note that the atMost parameter is now 30 minutes. This is because rebooting Windows for the first time after an installation takes extra time while automatic configuration is performed.

Sample Build Plan: Linux Scripted Install
Table 4 – ProLiant OS – RHEL 6.3 x64 Scripted Install build plan sample

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Parameters</th>
<th>Step Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check iLO Service</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>2</td>
<td>Boot</td>
<td>--serviceOS=linux64</td>
<td>OGFS</td>
</tr>
<tr>
<td>3</td>
<td>Decommission Server</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>4</td>
<td>Wait for HP SA Agent</td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td>OGFS</td>
</tr>
<tr>
<td>5</td>
<td>Set Media Source</td>
<td>@__OPSW-Media-LinURI@/rhel63-x64</td>
<td>Python</td>
</tr>
<tr>
<td>6</td>
<td>RHEL 6.3 x64 en_us Kickstart</td>
<td>/tmp/user.ks.cfg</td>
<td>Deploy configuration file</td>
</tr>
<tr>
<td>Step</td>
<td>Task Description</td>
<td>Platform</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Inject Required Kickstart Settings</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Red Hat Enterprise Linux Server 6 x86_64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Inject Kickstart Personalization Settings</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Create Stub Partition</td>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Copy Boot Media</td>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ProLiant Drivers for RHEL 6.3 x64</td>
<td>Deploy package</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>GRuB Boot Loader x86</td>
<td>Deploy package</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Deploy Agent</td>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-d /tmp/opt/opsware/agent/ogfs-agent.zip -u</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Embed files initrd</td>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-s /tmp/user.ks.cfg:/ -s /tmp/opt/opsware/agent:/opt/opsware/ -s /tmp/dud/.:/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Install bootloader for RedHat Enterprise Linux Server</td>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--kernel_arguments=&quot;@kernel_arguments@&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Reboot</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Wait for HP SA Agent</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Monitor Installation</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tmp/anaconda.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Integrate Linux HP SA Agent</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Reboot</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Wait for HP SA Agent</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--production --atLeast=3 --atMost=30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 1 – Early error detection**

The first step helps catch errors that might affect the running of the build plan later on. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the Steps 1 and 2 – Early error detection section.

**Steps 2 to 4 – Boot the server for provisioning**

The next set of steps are used to boot the server into the required service OS and rest the status of the server such that it can be provisioned. Once these steps are done, the server is in maintenance ready to start the provisioning process. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install sample and the detailed description in the...
**Steps 3 to 5 – Boot the server for provisioning** section. For the Linux build plans, the desired service OS **Boot** step parameter is **linux64**.

**Steps 5 to 15 – Stage the installation**

These steps perform all the work of setting up for the installation by copying files, putting drivers in place, and preparing the kickstart file.

- **Set Media Source** – This is the step that points the build plan to the OS distribution media on the media server. The parameter for this step is made up of a special custom attribute. This is a hidden custom attribute that corresponds to the information specified in the Media Server Settings page. At run time, this custom attribute is substituted and takes this form:

  
  http://media-server-IP/mnt/sharename/media

  where /mnt/sharename/media is the mount point and location to the OS distribution media.

- **RHEL 6.3 x64 en_us Kickstart** – This is a deploy configuration file step. It writes the HP-provided kickstart file to the target server’s ram drive. It is recommended to replace this step with a customized kickstart file using the HP-provided configuration files as a template since these template files contain the Custom Attribute syntax. The path specified by the parameter is where the file is written. Follow on build plan steps expect to find this file in that location, so it is not recommended that the path be changed.

- **Inject Required Kickstart Settings** – This is a script that adds required items to the kickstart file to make it work properly with the appliance. This script is required and always comes right after the step that writes out the kickstart file.

- **Inject Kickstart Personalization Settings** – This step is used when static IP addressing information is specified as part of the installation in the Run OS Build Plan page. Setting network information causes a server-level custom attribute named, hpsa_netconfig, to be created and assigned to each server being provisioned. The **Inject Kickstart Personalization Settings** step checks for the existence of that custom attribute. If it exists, the network information is read and the unattend file is modified to include the static IP information. This step should always follow the **Inject Required Kickstart Settings** step. Note: The hpsa_netconfig custom attribute is not removed after it is set, so it could be there if another build plan is run against the same target server.

- **Create Stub Partition** – Creates partition on the local disk to load the Linux build image.

- **Copy Boot Media** – Copies several files to the stub partition in order to boot the installer environment.

- **ProLiant Drivers for RHEL 6.3 x64** – This is a zip file containing the latest ProLiant drivers for this operating system. They are placed in the directory specified by the parameter, which is where the Linux OS installer is expecting to find them.

- **GRuB Boot Loader x86** – This is a zip file contain the grub boot loader that will land on the stub partition. Even though the name contains x86, it is still used with x64 Linux deployments.

- **Deploy Agent** – Copies the zip file containing the agent to the target server. This agent is used to communicate with the target server during the operating system installation and is not the same agent that is placed on the production server.

- **Embed files initrd** – Adds additional files to the new initrd image landed during the **GRuB Boot Loader x86** step.

- **Install bootloader for RedHat Enterprise Linux Server** – Installs the grub bootloader onto the stub partition in order to enable booting the Red Hat Enterprise Linux Server installation image. It takes the kernel_arguments optional custom attribute as a parameter to allow for additional kernel arguments to the installation kernel.

**Steps 16 to 19 – Perform the installation and post-install tasks**

These final steps actually perform the OS installation, add the production agent, and perform the final reboot of the server.

- **Reboot** – Reboots the target server for the purpose of booting the RHEL6.3 grub image and begin the Red Hat installation.

- **Wait for HP SA Agent** – This is the same step as before, but this time it is waiting for the agent that was deployed in build plan step 13 to register with the appliance.

- **Monitor Installation** - Periodically examines the operating system installation log file to check if the installation has completed.
• **Integrate Linux HP SA Agent** – This step adds the HP Server Automation agent to the newly installed operating system.

• **Reboot** – Initiates a reboot of the server to finalize the installation. Note that the Reboot step is used here instead of the Boot step. The Reboot step is specifically meant for rebooting a server to its local disk and back into production.

• **Wait for HP SA Agent** – This is the same step as before, but this time it is waiting for the agent to register in production mode. Also, note that the `atMost` parameter is now 30 minutes since rebooting Linux for the first time after an installation takes extra time.

---

**Sample Build Plan: ESXi Scripted Install**

*Table 5 – ProLiant OS – ESXi 5.1 Scripted Install build plan sample*

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Parameters</th>
<th>Step Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Validate Gateway Setting for Static Network Configuration</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Check iLO Service</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Boot</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--serviceOS=linux64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Decommission Server</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Wait for HP SA Agent</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Set Media Source</td>
<td>Python</td>
<td></td>
</tr>
<tr>
<td></td>
<td>@__OPSW-Media-LinURI@/esxi51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ESXi 5.1 Kickstart</td>
<td>Deploy configuration file</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/tmp/user.ks.cfg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Inject Required ESXi 5 Kickstart Settings</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--accept-encrypted-password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Inject Kickstart Personalization Settings for ESXi 5</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Create Stub Partition</td>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Copy Boot Media</td>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ESXi Installation Utilities</td>
<td>Deploy package</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Deploy Agent</td>
<td>Unix</td>
<td></td>
</tr>
</tbody>
</table>
21

-d /tmp/opt/opsware/agent/ogfs-agent.zip –p "Red Hat Enterprise Linux Server 5 X86_64" -u

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Add ESXi Module Unix</td>
<td>-s /opt/hpsa_agent -d</td>
</tr>
<tr>
<td>15</td>
<td>Add ESXi Module Unix</td>
<td>-s /tmp/user.ks.cfg -a ks.cfg</td>
</tr>
<tr>
<td>16</td>
<td>Install bootloader for ESXi Unix</td>
<td>--kernel_arguments=&quot;@kernel_arguments@&quot;</td>
</tr>
<tr>
<td>17</td>
<td>Reboot OGFS</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Wait for ESXi installation OGFS</td>
<td>--atLeast=3 --atMost=60</td>
</tr>
</tbody>
</table>

**NOTE:** For ESXi 5.1 and later, ESXi naming has changed to vSphere; however, OSBPs, scripts and configuration files may continue to state ESXi.

**Step 1 to 2 – Early error detection**

The first step helps catch errors that might affect the running of the build plan later on. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the *Steps 1 and 2 – Early error detection* section.

**Steps 3 to 5 – Boot the server for provisioning**

The next set of steps are used to boot the server into the required service OS and rest the status of the server such that it can be provisioned. Once these steps are done, the server is in maintenance ready to start the provisioning process. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install sample and the detailed description in the *Steps 3 to 5 – Boot the server for provisioning* section. For the ESXi build plans, the desired service OS Boot step parameter is linux64.

**Steps 6 to 16 – Stage the installation**

These steps perform all the work of setting up for the installation by partitioning the system drive, copying files, putting drivers in place, and preparing the kickstart file.

- **Set Media Source** – This is the step that points the build plan to the OS distribution media on the media server. The parameter for this step is made up of a special custom attribute. This is a hidden custom attribute that corresponds to the information specified in the Media Server Settings page. At runtime, this custom attribute is substituted and takes this form:
  ```
  http://media-server-IP/mnt/sharename/media
  ```
  where `/mnt/sharename/media` is the mount point and location to the OS distribution media.

- **ESXi 5.1 Kickstart** – This is a deploy configuration file step. It writes the HP-provided kickstart file to the target server’s ram drive. It is recommended to replace this step with a customized kickstart file using the HP-provided configuration files as a template since these template files contain the Custom Attribute syntax. The path specified by the parameter is where the file is written. Follow on build plan steps expect to find this file in that location, so it is not recommended that the path be changed.

- **Inject Required ESXi 5 Kickstart Settings** – This is a script that adds required items to the kickstart file to make it work properly with the appliance. This script is required and always comes right after the step that writes out the kickstart file.

- **Inject Kickstart Personalization Settings for ESXi 5** – This step is used when static IP addressing information is specified as part of the installation in the Run OS Build Plan page. Setting network information there causes a server-level custom attribute named, hpsa_netconfig, to be created and assigned to each server being provisioned. The *Inject Kickstart Personalization Settings for ESXi 5* step checks for the existence of that custom attribute. If it exists, the network information is read and the unattend file is modified to include the static IP information. This step should always follow the *Inject Required ESXi 5 Kickstart Settings* step. Note:
The hpsa_netconfig custom attribute is not removed after it is set, so it could be there if another build plan is run against the same target server.

- **Create Stub Partition** – Creates partition on the local disk to load the Linux build image.
- **Copy Boot Media** – Copies several files to the stub partition in order to boot the installer environment.
- **ESXi Installation Utilities** - A package containing the boot loader and Master Boot Record needed to boot ESXi.
- **Deploy Agent** – Refer to Table 4 – ProLiant OS – RHEL 6.3 x64 Scripted Install build plan sample and the detailed description in the *Steps 5 to 15 – Stage the installation* section.
- **Add ESXi Module** – Places the file or directory specified as an argument to the “-s” into a compressed tar file so that it can be loaded as an ESXi module when ESXi is booted. The file or directory will be visible in the file system when ESXi boots.
- **Install bootloader for ESXi** – Installs the grub bootloader onto the stub partition in order to enable booting the ESXi installation image. It takes the kernel_arguments optional custom attribute as a parameter to allow for additional kernel arguments to the installation kernel.

**Steps 17 and 18 – Perform the installation and post-install tasks**

These final steps actually perform the OS installation, add the production agent, and perform the final reboot of the server.

- **Reboot** – Reboots the target server which on next boot will boot the ESXi 5.1 grub image and begin the ESXi installation.
- **Wait for ESXi Installation** – Since there is no SA agent support for ESXi operating system, this step is special compared to the **Wait for HP SA Agent** step and detects when the ESXi installation is complete and running the production operating system.

### Sample Build Plan: ProLiant Hardware Configuration

**Table 6 – ProLiant HW – System ROM Enable Boot from SAN on Bladeserver build plan sample**

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Type</th>
<th>Step Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check iLO Service</td>
<td>OGFS</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Boot</td>
<td>OGFS</td>
<td>--serviceOS=linux64</td>
</tr>
<tr>
<td>3</td>
<td>Wait for HP SA Agent</td>
<td>OGFS</td>
<td>--maintenance --atLeast=3 --atMost=20</td>
</tr>
<tr>
<td>4</td>
<td>LinuxPE add-on packages</td>
<td>Deploy package</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ProLiant Scripting Toolkit for Linux x64</td>
<td>Deploy package</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>System ROM Configuration – Enable Boot from SAN</td>
<td>Deploy configuration file</td>
<td>/tmp/sstk/conrep-bfs.xml</td>
</tr>
<tr>
<td>7</td>
<td>Manage System Configuration</td>
<td>Unix</td>
<td>-l /tmp/sstk/conrep-bfs.xml</td>
</tr>
<tr>
<td>8</td>
<td>Boot</td>
<td>OGFS</td>
<td></td>
</tr>
</tbody>
</table>
Step 1 – Early error detection
The first step helps catch errors that might affect the running of the build plan later on. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the Steps 1 and 2 – Early error detection section.

Steps 2 to 3 – Boot the server for provisioning
The next set of steps are used to boot the server into the required service OS and rest the status of the server such that it can be Provisioned. Once these steps are done, the server is in maintenance ready to start the provisioning process. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install sample and the detailed description in the Steps 3 to 5 – Boot the server for provisioning section. For the hardware build plans, the desired service OS Boot step parameter is linux64.

Steps 4 to 7 – Apply the hardware configuration
These steps perform all the work of setting up for and configuring the target server environment, by landing appropriate ProLiant Scripting Toolkit files and libraries and configuration file and running the tool to apply the configuration.

• **LinuxPE add-on packages** – This is a zip file containing additional libraries and utilities required in the LinuxPE PXE service OS.

• **ProLiant Scripting Toolkit for Linux x64** – This is a zip file containing the ProLiant Scripting Toolkit utilities.

• **System ROM Configuration – Enable Boot from SAN** – This is a deploy configuration file step. It writes the HP-provided hardware configuration file to the target server’s ram drive. The configuration file format aligns with the ProLiant Scripting Toolkit utility requirements. The path specified by the parameter is where the file is written. Follow on build plan steps expect to find this file in that location, so it is not recommended that the path be changed.

• **Manage System Configuration** – This step actually runs the appropriate ProLiant Scripting Toolkit utility using the configuration file as a parameter.

Steps 8 to 9 – Perform the post configuration tasks
These final steps perform the final reboot of the target server to apply the new configuration.

• **Boot** – Reboots the target server back into the service OS to force the ProLiant Scripting Toolkit utility changes to be available at next reboot.

• **Wait for HP SA Agent** – This is the same step as before, but this time it is waiting for the agent to register in production mode.

Sample Build Plan: ProLiant Software Install
Table 7 – ProLiant SW – Offline Firmware Update build plan sample

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Parameters</th>
<th>Step Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check iLO Service</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>2</td>
<td>Boot</td>
<td>--serviceOS=linux64</td>
<td>OGFS</td>
</tr>
<tr>
<td>3</td>
<td>Wait for HP SA Agent</td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td>OGFS</td>
</tr>
</tbody>
</table>
Step 1 – Early error detection
The first step helps catch errors that might affect the running of the build plan later on. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install build plan sample and the detailed description in the Steps 1 and 2 – Early error detection section.

Steps 2 and 3 – Boot the server for provisioning
The next set of steps are used to boot the server into the required service OS and rest the status of the server such that it can be provisioned. Once these steps are done, the server is in maintenance ready to start the provisioning process. Refer to Table 1 – ProLiant OS – Windows 2012 Standard x64 Scripted Install sample and the detailed description in the Steps 3 to 5 – Boot the server for provisioning section. For the hardware build plans, the desired service OS Boot step parameter is linux64.

Steps 4 and 5 – Prepare for firmware update
These steps perform the work of setting up for firmware update on the target server by connecting to the Media Server where the SPP files are located and landing appropriate libraries needed in the service OS environment.

- **Set Media Source** – This is the step that points the build plan to the SPP bundle on the Media Server. The parameter for this step is made up of three special custom attributes all strung together. These are hidden custom attributes that correspond to the information specified in the Media Server Settings page. These custom attributes should always be used together. At run time, these custom attributes are substituted, and the result is a URI that takes this form:
  
  smb://username:password@media-server-IP/share-name#mount_point?noserverino

  where *username:password* are the credentials for the media server, and *mount_point* is the share that will be mounted. Optionally, all three custom attributes can be replaced with customized values as long as it conforms to URI specification above. Note that the *noserverino* option is necessary when mounting a Windows share using the Linux service OS Samba client.

- **LinuxPE add-on packages** – Refer to Table 6 – ProLiant HW – System ROM Configuration Boot from SAN build plan sample and the detailed description in the Steps 4 to 6 – Prepare for hardware configuration section.

Steps 6 to 8 – Perform the update and post-install tasks
These final steps actually perform the firmware update and perform a reboot of the server.

- **Update Firmware Using SPP** – This step actually runs the HP Smart Update Manager to upgrade the firmware on the target server. Since the Media Server can contain several SPP versions, this script takes a parameter with a SPPversion custom attribute name to assign a specific SPP version or it'll use the latest in the Media Server. This script expects the SPP location on the Media Server to be `/Media/SPP/yyyy.xx` where `yyyy.xx` is the SPP version number. This path cannot be easily changed.

- **Boot** – Reboots the target server back into the service OS to force the ProLiant Scripting Toolkit utility changes to be available at next reboot.
- **Wait for HP SA Agent** – This is the same step as before, but this time it is waiting for the agent to register in production mode.

**Sample Build Plan: Combination Hardware, Windows Scripted Install, and SPP**

Table 8 – ProLiant COMBO - BFS + Windows 2012 + SPP build plan sample

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Step Name</th>
<th>Step Parameters</th>
<th>Step Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Validate Custom Attributes</td>
<td>--custAttrNames &quot;ProductKey_Win2012-Std-x64&quot;</td>
<td>OGFS</td>
</tr>
<tr>
<td>2</td>
<td>Check iLO Service</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>3</td>
<td>Boot</td>
<td>--serviceOS=winpe64</td>
<td>OGFS</td>
</tr>
<tr>
<td>4</td>
<td>Decommission Server</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>5</td>
<td>Wait for HP SA Agent</td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td>OGFS</td>
</tr>
<tr>
<td>6</td>
<td>LinuxPE add-on packages</td>
<td></td>
<td>Install ZIP</td>
</tr>
<tr>
<td>7</td>
<td>ProLiant Scripting Toolkit for Linux x64</td>
<td></td>
<td>Deploy package</td>
</tr>
<tr>
<td>8</td>
<td>System ROM Configuration – Enable Boot from SAN</td>
<td>/tmp/sstk/conrep-bfs.xml</td>
<td>Deploy configuration file</td>
</tr>
<tr>
<td>9</td>
<td>Manage System Configuration</td>
<td>-l /tmp/sstk/conrep-bfs.xml</td>
<td>Unix</td>
</tr>
<tr>
<td>10</td>
<td>Boot</td>
<td>--serviceOS=winpe64</td>
<td>OGFS</td>
</tr>
<tr>
<td>11</td>
<td>Wait for HP SA Agent</td>
<td>--maintenance --atLeast=3 --atMost=20</td>
<td>OGFS</td>
</tr>
<tr>
<td>12</td>
<td>Set Media Source</td>
<td>@__OPSW-Media-WinUser@@__OPSW-Media-WinPassword@@__OPSW-Media-WinPath#@z</td>
<td>Python</td>
</tr>
<tr>
<td>13</td>
<td>Prepare Disks on HP ProLiant Gen8</td>
<td></td>
<td>OGFS</td>
</tr>
<tr>
<td>14</td>
<td>Create Windows System Drive</td>
<td></td>
<td>Windows .BAT</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Windows 2012 Standard x64 en_us Unattend Deploy configuration file x:\Windows\Temp\Unattend.xml</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Inject Required Unattend.xml Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Inject Personalization Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>ProLiant Drivers for Windows 2012 -- yyyymm Install ZIP @SystemDrive:c$oem$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Run Windows 2012 x64 Setup &quot;z:\Media\win2012-x64-en_us\setup.exe&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Integrate HP SA Agent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Reboot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Wait for HP SA Agent --production --atLeast=3 --atMost=30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Set Media Source Python @__OPSW-Media-WinUser@@__OPSW-Media-WinPassword@@__OPSW-Media-WinPath#@/mnt/ms?noserverino</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Get Deployment Interface Details Python @systemDrive:c\ServerDetails.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Install Windows SPP Windows .BAT --spp_version=@SPPversion:latest@</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Wait for HP SA Agent --production --atLeast=1 --atMost=30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Reboot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Wait for HP SA Agent --production --atLeast=3 --atMost=30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**How these build plans were combined**

This COMBO build plan gives an example of how multiple individual build plans that are frequently used together can be combined into one large build plan. The steps themselves will not be discussed in this section as they have been discussed in previous sections. Instead, this section will go into detail about what was done to combine these build plans and what modifications were made to the build plan once they were combined. Use the information in this section as a guide when creating other combined build plans.

**What this build plan does**

This build plan is an example of combining a set of actions that frequently go together; enabling the boot from SAN functionality, installing the operating system, and installing the SPP. It is a combination of three of the build plans that are provided with the appliance.
- ProLiant HW - System ROM Enable Boot from SAN on Bladeserver
- ProLiant OS - Windows 2012 Standard x64 Scripted Install
- ProLiant SW - Install Windows SPP

It would have been acceptable to take these three build plans and simply concatenate the steps and parameters in order to make this combined build plan, but several modifications were made to make the build plan more efficient and more reliable. It is these modifications that are described in the rest of this section.

The descriptions below refer to the three sections of the build plan. These sections correspond to the groupings of steps that make up each of the three concatenated build plans.

**Step 1 – Move error checking to head of build plan**
The *Validate Custom Attributes* build plan script step was moved from the beginning of the OS installation section to the beginning of the combined build plan. Remember, this step is used for early error detection. It is better to do all the error checking at the head of the build plan, so if the custom attribute is not defined, the build plan will fail with this error before the system ROM is modified.

**Step 4 – Decommission Server after the first Boot step**
The *Decommission Server* build plan script step was moved from the OS installation section to after the first *Boot* build plan script step of the combined build plan. As described above, the *Decommission Server* step is used when reprovisioning a server that already has an OS installed on it. When using this step, it must always be used after the first *Boot* step of the build plan or the target server may not be able to boot into the service OS to perform the OS installation.

NOTE: Moving this step is mandatory if you plan on reprovisioning a server with this build plan.

**Step 10 – Modified Boot step service OS**
All hardware configuration build plans run in the Linux service OS, so in the original Boot from SAN hardware configuration build plan, the step that reboots the server to apply the settings will boot the server back into the Linux service OS. In this build plan, however, we are about to install Windows, so this boot step was modified to reboot the server into the WinPE service OS. Had this step not been modified, an additional *Boot* step would have been needed later to switch to the WinPE service OS which would make the build plan run that much longer while the server was rebooted.

**Step 11 – Last step of the Boot from SAN section**
The *Wait for SA Agent* build plan script step will wait for the agent in the WinPE service OS before beginning the OS installation.

**Before Step 12 – Beginning of OS installation section - Unnecessary steps removed**
Step 12 begins the OS installation section of the build plan. The first five steps of the original OS installation build plan have been moved or removed.

- *Validate Custom Attributes* was moved to the head of the build plan.
- *Check iLO Service* was already done at the head of the build plan so it was removed.
- The *Boot* step was removed since the target server will already be in the WinPE service OS as a result of modifying Step 10 *Boot* step. Refer to Step 10 – Modified *Boot* step service OS section above.
- *Decommission Server* was moved to be after the first *Boot* step of the build plan.
- *Wait for HP SA Agent* is not needed since the *Boot* step was removed.

**Steps 12 to 22 – The rest of the OS Installation build plan**
These are the steps that perform the actual OS installation and are not modified from the original build plan.

**Before Step 23 – Remove unnecessary steps**
The *Check iLO Service* step was removed from the SPP section as that step was already performed at the beginning of the combined build plan.

**Steps 23 to 28 – The rest of the SPP build plan**
These are the steps that perform the actual SPP installation and are not modified from the original build plan.
Build Plan and Build Plan Steps Reference

OS Build Plans

Summary
The build plans listed in this section are the HP-provided build plans.

ProLiant COMBO - BFS + Windows 2012 + SPP
This build plan configures ProLiant Bladeservers’ System ROM to enable Boot from SAN functionality (BFS), performs a scripted install of Windows 2012 Standard using a generic unattend file, and installs the Service Pack for ProLiant (SPP) on the Windows operating system.

Requirements:
- HP ProLiant Bladeserver with iLO and fibre channel controller
- IC server provisioning Media Server must contain the OS distribution
- IC server provisioning Media Server must contain a SPP

Required Custom Attributes:
- ProductKey_Win2012-Std-x64 set using the Settings > Product Key page

Optional Custom Attributes:
- ComputerName
- AdminPassword (must be encrypted)
- SystemDisk
- SystemDiskNumber

ProLiant COMBO - BFS + RHEL6.4 + SPP
This build plan configures ProLiant Bladeservers’ System ROM to enable Boot from SAN functionality (BFS), performs a scripted install of Red Hat Enterprise Linux 6.4 x64 using a generic kickstart file, and installs the Service Pack for ProLiant (SPP) on the Linux operating system.

Requirements:
- HP ProLiant Bladeserver with iLO and fibre channel controller
- IC server provisioning Media Server must contain the OS distribution
- IC server provisioning Media Server must contain a SPP

Required Custom Attributes:
- ProductKey_Win2012-Std-x64 set using the Settings > Product Key page where xxx is the Windows version and yyy is the architecture version.

Optional Custom Attributes:
- root_password (must be encrypted)
- boot_disk
- kernel_arguments

ProLiant HW - Add Migrated Linux Server
ProLiant HW - Add Migrated Windows Server
Automatically registers an RDP migrated server’s iLO by temporarily rebooting the server into maintenance.

NOTE: These two build plans are designed to work only with target servers that were migrated to this appliance from RDP. They make use of software that is installed as part of an RDP installation.
NOTE: These build plans will reboot the running target server into maintenance.

When a target server is initially migrated to IC server provisioning from RDP, the iLO from that server is still unknown to the appliance. Since the iLO is required for most build plans, the iLO must be registered before other build plans can be run. This can be done manually, or automatically by using these build plans.

Add Migrated Linux Server uses the /sbin/hpbootcfg utility to set the one-time PXE boot to a service OS and is run on Linux-based targets. Add Migrated Windows Server uses the ProLiant Scripting Toolkit hponcfg utility to set the one-time PXE boot to a service OS and is run on Windows-based targets. The target server will be booted into the LinuxPE environment where the server's iLO will be registered and then booted back into production.

Requirements:
- Runs only on ProLiant Servers with iLO that were migrated from an RDP server.
- Appliance must have DHCP/PXE support.
- Presence of /sbin/hpbootcfg for Linux-based servers.

Custom Attributes: None

**ProLiant HW - Boot Linux Service OS**
**ProLiant HW - Boot WinPE Service OS**

Boots the target server into the appropriate service OS, maintenance mode, per the build plan title.

These are the build plans used when adding a server via its iLO and booting to maintenance.

Requirements: HP ProLiant server with iLO

Custom Attributes: None

**ProLiant HW - Boot Local Disk**

Boots a target server into production via its local disk.

NOTE: The build plan expects an agent to be running on the target when the build plan completes or the build plan will fail, even if the target boots successfully.

Requirements: HP ProLiant server with iLO

Custom Attributes: None

**ProLiant HW - Erase Server**

Resets a server and disks to factory defaults.

IMPORTANT: Running this build plan causes data loss.

This build plan deletes the partition table on the server's disks, clears the Smart Array configuration on any attached Smart Array controllers, and resets the system BIOS settings to default values. The server's state is changed to unmanaged. Note that when the BIOS is reset, the BIOS date/time becomes incorrect.

Requirements: HP ProLiant server with iLO

Custom Attributes: None

**ProLiant HW - Fibre Channel HBA Configure Boot Device**

Configures the boot device for an Emulex HBA, Emulex CNA, or Qlogic HBA using custom attributes.

To support Emulex or QLogic HBA, or Emulex CNA scripted configuration, this build plan configures the boot device using the settings specified in the HBA_Config custom attribute. The configuration of the boot device is necessary in order to allow the server to boot from SAN. For further information on how this build plan is used, refer to the How Do I...? section of the IC server provisioning online help.

Requirements: HP ProLiant server with iLO and an Emulex HBA, Emulex CNA or QLogic HBA but not Emulex and QLogic at the same time. 
Required Custom Attributes: HBA.Config

ProLiant HW - Fibre Channel HBA Display Configuration
Displays the Emulex or QLogic HBA, or Emulex CNA configuration settings to the job log.

The Emulex or QLogic HBA, or Emulex CNA configuration is displayed to provide information about any HBAs found on the target server, such as the HBA WWPN, link status, enable BIOS setting, selectable boot setting, primary boot port name, and LUN.

Requirements: HP ProLiant server with iLO and an Emulex HBA, Emulex CNA or QLogic HBA but not Emulex and QLogic at the same time.

Custom Attributes: None

ProLiant HW - iLO Capture Configuration
Captures HP Integrated Lights-Out settings into a configuration file. The configuration file is stored on the target server and then uploaded to the appliance.

The configuration file format is defined by the ProLiant Scripting Toolkit iLO utility.

Requirements: HP ProLiant server with iLO

Custom Attribute: configuration_location – The absolute filename of the captured configuration xml on the target server.

ProLiant HW - iLO Set Minimum Password Length
Sets the HP Integrated Lights-Out minimum password length.

This is a sample build plan that can be modified to set any iLO settings. The configuration file format is defined by the ProLiant Scripting Toolkit iLO utility.

Requirements: HP ProLiant server with iLO

Custom Attributes: None

ProLiant HW - Prepare for Linux Reprovisioning
ProLiant HW - Prepare for Windows Reprovisioning
Prepares a provisioned target server for reprovisioning by decommissioning it and booting it into a service OS (maintenance mode).

Target servers provisioned by IC server provisioning contain certificates that uniquely and securely identify the server to the appliance. The appliance expects to retrieve this information every time the target server boots. This can cause problems when attempting to reprovision a server or when a server’s hard drive is erased, as the service OS may not be able to retrieve these certificates. These build plans contain the Decommission Server step, which breaks that security requirement as part of the process of booting into the service OS.

NOTE: Only run this if you are certain that you will not want the target server to boot into production mode any more. Once you run this build plan on a server, the production agent on that server will not be able to communicate with the appliance.

Requirements: HP ProLiant server with iLO

Custom Attributes: None

ProLiant HW - Smart Array Capture Settings
Captures the HP Smart Array current configuration into a configuration file. The configuration file is stored on the target server and then uploaded to the appliance.

The configuration file format is defined by the HP ACU scripting utility and may be used as input for Manage Smart Array Configuration script. This build plan will unmounts the production drives while in the service OS.

Requirements: HP ProLiant server with iLO and a Smart Array controller
Custom Attribute: configuration_location – The absolute filename of the captured configuration xml on the target server.

**ProLiant HW - Smart Array Erase**
This build plan resets the Smart Array, clearing the configuration.
IMPORTANT: Running this build plan causes the loss of all data on the Smart Array disks.
The server's state will change to unmanaged, due to potential loss of critical identification and agent association data.
Requirements: HP ProLiant server with iLO and a Smart Array controller
Custom Attributes: None

**ProLiant HW - Smart Array Set RAID 1**
Sets the HP Smart Array configuration for RAID 1 using two drives.
This is a sample build plan that can be modified to set any Smart Array settings. The configuration file format is defined by the ProLiant Scripting Toolkit ACU scripting utility.
IMPORTANT: Any existing configuration is cleared, resulting in loss of existing data on the Smart Array. Additionally, the server's state will change to unmanaged, due to potential loss of critical identification and agent association data.
Requirements: HP ProLiant server with iLO and a Smart Array controller
Custom Attributes: None

**ProLiant HW - System ROM Capture Settings**
Captures the ProLiant System ROM settings into a configuration file. The configuration file is stored on the target server and then uploaded to the appliance.
The configuration file format is defined by the HP System ROM scripting utility and may be used as input for the Manage System ROM Configuration script.
Requirements: HP ProLiant server with iLO
Custom Attribute: configuration_location – The absolute filename of the captured configuration xml on the target server.

**ProLiant HW - System ROM Enable Boot from SAN on Bladeserver**
Configures the ProLiant BladeSystem ROM to enable Boot from SAN functionality.
Boot from SAN is enabled by setting the fibre channel first in the boot order and disabling any embedded storage controller.
Requirements: HP ProLiant BladeSystem with iLO and fibre channel controller
Custom Attributes: None

**ProLiant OS**
- **ESXi 5.0 U1 Scripted Install**
- **ESXi 5.0 U2 Scripted Install**
- **ESXi 5.1 Scripted Install**
- **ESXi 5.1 U1 Scripted Install**
- **ESXi 5.5 Scripted Install**
Performs a scripted install of the selected VMware ESXi version using a generic kickstart file.
NOTE: For ESXi 5.1 and later, ESXi naming has changed to vSphere; however, OSBPs, scripts and configuration files may continue to state ESXi.
Requirements:
  - HP ProLiant server with iLO
IC server provisioning Media Server must contain the OS distribution

Optional Custom Attributes:
- ProductKey_ESXiZZ corresponding to the selected version of ESXi where ZZ is the version number
- root_password (must be encrypted)
- boot_disk
- kernel_arguments

ProLiant OS - RHEL 5.9 x64 Scripted Install
ProLiant OS - RHEL 6.3 x64 Scripted Install
ProLiant OS - RHEL 6.4 x64 Scripted Install
ProLiant OS - RHEL 6.4 x64 KVM Scripted Install

Performs a scripted install of the selected Red Hat Enterprise Linux version using a generic kickstart file.

The ProLiant OS – RHEL 6.4 x64 KVM Scripted Install build plan installs a RHEL6.4 x64 Kernel Virtual Machine (KVM) hypervisor on a target server, known as the KVM host. Once the KVM hypervisor is installed, libvirt utilities such as virsh or virt-manager may be used to add KVM guests on the KVM host.

Additionally, virtualization must be enabled in the BIOS. Virtualization is enabled by default; however, if it has been disabled on the target server, it can be enabled in the RBSU under the System Options -> Processor Options menu, or create a build plan using the ProLiant HW - System ROM Enable Boot from SAN on Bladeserver build plan, replacing the configuration file with the System ROM Configuration – Enable Virtualization configuration file.

Requirements:
- HP ProLiant server with iLO
- IC server provisioning Media Server must contain the OS distribution

Optional Custom Attributes:
- root_password (must be encrypted)
- boot_disk
- kernel_arguments

ProLiant OS - SLES11 SP2 x64 Scripted Install
ProLiant OS - SLES11 SP3 x64 Scripted Install

Performs a scripted install of the selected SUSE Enterprise Linux version using a generic autoyast file.

NOTE: To install SUSE on some newer ProLiant Gen8 servers, special kISO bootloader support is required. Refer to the IC server provisioning On-line Help.

Requirements:
- HP ProLiant server with iLO
- IC server provisioning Media Server must contain the OS distribution

Optional Custom Attributes:
- root_password (must be encrypted)
- boot_disk
- kernel_arguments

ProLiant OS - Windows 2008 SP2 Standard x64 Image Capture
ProLiant OS - Windows 2008 R2 SP1 Standard x64 Image Capture
ProLiant OS - Windows 2012 Standard x64 Image Capture

Captures the selected version of Windows into a WIM image.
The WIM image can then be used to clone multiple target servers. Upon completion of the build plan, the reference server from which the image was captured is left in a depersonalized state.

Requirements:
- HP ProLiant server with iLO must be powered on and running the Windows OS to be captured.
- Media Server share must be writeable.
- A WinPE image has been generated by the WinPE image generation utility and uploaded to the appliance using Settings > DHCP page. Uploading a WinPE image to the appliance also adds the imagex.exe utility which is required for all image capture and deploy operations. The ImageX package shipped with the appliance does not contain the actual imagex.exe utility by default due to licensing restrictions.

Required Custom Attributes:
- WimFileName - The file name where the captured image will be stored on the Media Server.

**ProLiant OS - Windows 2008 SP2 Standard x64 Image Install**
**ProLiant OS - Windows 2008 R2 SP1 Standard x64 Image Install**
**ProLiant OS - Windows 2012 Standard x64 Image Install**
Applies a previously captured Windows WIM image. This build plan is used to clone servers using an image captured from a reference server.

Requirements:
- HP ProLiant server with iLO and similar hardware as the reference server.
- A previously captured WIM image on Media Server.
- A WinPE image has been generated by the WinPE image generation utility and uploaded to the appliance using Settings > DHCP page. Uploading a WinPE image to the appliance also adds the imagex.exe utility which is required for all image capture and deploy operations. The ImageX package shipped with the appliance does not contain the actual imagex.exe utility by default due to licensing restrictions.

Required Custom Attributes:
- ProductKey_xxx-yyyy corresponding to the selected version of Windows, set using the Settings > Product Key page where xxx is the Windows version and yyy is the architecture version.
- WimFileName - A previously captured image file name

Optional Custom Attributes:
- ComputerName
- AdminPassword (must be encrypted)
- SystemDisk
- SystemDiskNumber

**ProLiant OS - Windows 2008 SP2 Standard x64 Scripted Install**
**ProLiant OS - Windows 2008 R2 SP1 Standard x64 Scripted Install**
**ProLiant OS - Windows 2012 Standard x64 Scripted Install**
Performs a scripted install of the selected Windows version using a generic unattend file.

Requirements:
- HP ProLiant server with iLO
- IC server provisioning Media Server must contain the OS distribution

Required Custom Attributes:
- ProductKey_xxx-yyyy corresponding to the selected version of Windows, set using the Settings > Product Key page where xxx is the Windows version and yyy is the architecture version.
Optional Custom Attributes:

- ComputerName
- AdminPassword (must be encrypted)
- SystemDisk
- SystemDiskNumber

**ProLiant SW - Install Linux SPP**

**ProLiant SW - Install Windows SPP**

Installs the Service Pack for ProLiant (SPP) on a Windows or Linux production target server.

By default, the entire SPP will be installed on the server, including firmware. Options can be passed to the Install Linux SPP or Install Windows SPP step that can control what gets installed. See the HP SUM User Guide for information about what options are available.

Requirements:

- HP ProLiant server with iLO and running a production operating system
- IC server provisioning Media Server must contain a SPP

Custom Attributes: None

**ProLiant SW - Offline Firmware Update**

Updates the firmware using the Service Pack for ProLiant (SPP).

The target server will be booted into the Linux service OS and the SPP firmware update function will be run. Upon completion of this build plan, the target server will be left in the service OS. If you require the production OS, the build plan will need to be modified with the appropriate boot steps.

Requirements:

- HP ProLiant server with iLO
- IC server provisioning Media Server must contain a SPP

Optional Custom Attributes:

- SPPversion

**Scripts**

**Summary**

The scripts listed in this section are used or were created to be used by the HP-provided build plans. There are some scripts listed here that are not part of the HP-provided build plans, but are meant to provide extended capabilities when creating custom build plans.

NOTE: There are also some scripts on your appliance that are not listed in this document. These scripts are part of the underlying Server Automation installation. They have not been tested with the appliance and are not guaranteed to work.

**Add ESXi Module**

Adds the specified module to the list of modules ESXi will boot.

Places the file or directory specified as an argument to the “-s” into a compressed tar file so that it can be loaded as an ESXi module when ESXi is booted. The file or directory will be visible in the file system when ESXi boots.

Type: Unix

Parameters:

- -s file – a file or directory to create a module from
-d – create a tar in tar image to get around VMware’s vgz format
-a file – alias the file or directory

Custom Attributes: None

**Add iLO User**

Adds an iLO user using the ProLiant Scripting Toolkit hponcfg utility.

Type: OGFS

Required Parameters:
- --username=USERNAME where USERNAME of the new iLO user.
- --password=PASSWORD where PASSWORD of the new iLO user
- --privilege=PRIVILEGE where PRIVILEGE to be given to the new iLO user. Can be specified more than once.
  Possible values:  ADMIN REMOTE_CONS RESET_SERVER VIRTUAL_MEDIA CONFIG_ILO

Custom Attributes: None

**Add Windows Hyper-V Role**

**Add Windows Multipath IO Feature**

Installs the Windows Hyper-V Role or Multipath IO Feature using ServerManagerCmd or PowerShell.

These scripts can be run on a Windows production OS or appended to the end of a Windows installation build plan to add the specified feature/role. A reboot is required after this script is run.

These scripts can also be used as templates for enabling other roles.

Type: Windows BAT

Parameters: None

Custom Attributes: None

**Add Windows Server to Domain**

Adds a target server to a Domain.

This script can be run on a Windows production OS or appended to the end of a Windows installation build plan. A reboot is required after this script is run. This script will fail if the required custom attributes are not defined or if a DNS server is not configured on the target server.

NOTE: This script uses PowerShell commands. To run on Windows 2008, PowerShell 2.0 and WinRM need to be installed.

Type: Windows BAT

Requirements: DNS needs to be configured on the target server

Parameters: None

Required Custom Attributes:
- DomainFQDN – Fully Qualified Domain Name
- DomainName – NETBIOS name of domain
- DomainUser – domain user with permissions to join the domain
- Password, either
  - DomainPassword – text-based domain password to join the domain
  OR
  - EncryptedDomainPassword – encrypted domain password to join the domain, and
○ Key – Used in combination with EncryptedDomainPassword. Key used to generate the encrypted password.

**Adjust Windows System Disk Number on HP ProLiant Gen8**
Adjusts the SystemDiskNumber custom attribute on Gen8 servers to account for the Virtual Install Disk.

The diskpart command may list the Virtual Install Disk before the system disk on some Gen8 servers. When this situation occurs, the SystemDiskNumber custom attribute needs to be adjusted so that it references the correct system disk and not the Virtual Install Disk. The server’s SystemDiskNumber custom attribute is updated with the new value.

Type: OGFS

Optional Parameter:
- `--ca="Alternate-Custom-Attribute-Name"` Specifies an alternate custom attribute name where the adjusted system disk number will get saved. The default custom attribute is SystemDiskNumber.

Custom Attributes: SystemDiskNumber

**An Empty Template OGFS Script**
**An Empty Template Python Script**
**An Empty Template Unix Shell Script**
**An Empty Template Windows Batch Script**
**An Empty Template Windows VBScript**

These template scripts can be used to create new scripts of specified type. As of IC server provisioning release 7.2.2, these scripts have been deprecated since creating a new script is now available from the user interface.

Type: OGFS, Python, Unix, Windows .BAT, or Windows VBScript

Parameters: None

Custom Attributes: None

**Apply WIM Image**
Takes a previously captured Windows image WIM image that is saved on the Media Server and installs it on the target server’s boot disk.

Type: Windows .BAT

Parameters: None

Custom Attributes: None

**Boot**
Boots a target server into the specified service OS.

This Boot step is used in almost all build plans to put the target server into the appropriate service OS for running the rest of the steps in the build plan. It first checks the current state of the target server. If the server is already running the specified service OS, the boot step does nothing and exits with a success code. If a boot is required, the boot step will contact the target server’s iLO to power off the server, set the one time boot flag, and the power the server back on.

**IMPORTANT:**
- The boot step completes as soon as the boot is initiated, and does not wait for a successful boot. For this reason, the boot step should always be followed by the Wait for SA Agent step. Sometimes the Decommission Server step can come between Boot and Wait for SA Agent.
- When checking the current state of the target server, the boot step cannot distinguish between a server that was booted using PXE or one booted using the embedded service OS. If the build plan requires one specific context or the other, be sure to specify the `-force` option, which will cause a server reboot regardless of the current state.

Type: OGFS
Required Parameters:

- `--serviceOS=service_os` where `service_os` is the name of the service OS in which to boot. Possible values are `linux`, `linux64`, and `winpe64`. There is no difference between `linux` and `linux64`.

Optional Parameters:

- `--method=method_options` where `method_options` are
  - `auto` – The boot method will be chosen automatically based on the server type. This is the default.
  - `embedded` – Boots to the embedded service OS. This is supported only for HP ProLiant Gen8 and newer servers.
  - `network` – Boots the server using PXE, regardless of the type of server.

- `--force` - Forces a reboot of the server, regardless of the current state or service OS.

Custom Attributes: None

Capture Windows Image

Creates a WIM image of the target server using ImageX.

For more information about ImageX, see the following two articles:


Type: Windows .BAT

The parameters are specified in order, separated by spaces.

Required Parameters:

- Full path where the WIM image file will be saved including the file name and .wim extension.
- Drive to be captured

Optional Parameter:

- Path to the optional wimscript.ini exclusion file. This file contains a list of files for ImageX to skip while imaging. The Capture Windows Image script will always add files to the exclusion list to prevent imaging the Server Automation agent files. In no wimscript.ini file is specified, one will be created just for the agent files.

Example parameters:

- `Z:\media\newWimFile.wim C:\Z\configs\wimscript.ini`

Required Custom Attributes: None

Check iLO Service

Checks that the target server’s iLO has been registered properly with the appliance.

There are circumstances where a target server may appear in the appliance interface, but its iLO may not have completed registering itself. Since the Boot step requires the iLO for its boot control functions, a build plan could unnecessarily fail if the iLO registration process is still in progress. The `Check iLO Service` script is added at the head of most build plans before the `Boot` step to verify that the target server has a valid iLO registered. If no iLO is registered, the script will wait some amount of time to allow the iLO to complete its registration. Once the iLO becomes registered, the step completes and the build plan continues. If no iLO is found before the timeout, the script fails and halts the build plan.

Type: OGFS

Optional Parameter:

- `--atMost=minutes` where minutes is the time before timing out. Default is 10 minutes.

Custom Attributes: None
**Configure Fibre Channel HBA Boot Device**

Configures a QLogic or Emulex HBA, or Emulex CNA boot device by applying the configuration specified in the HBA_Config multi-line custom attribute.

**Type:** Python

**Optional Parameter:**
- `--displayHbaOnly` - Show the HBA or CNAs on the target server, but doesn't apply the configuration.

**Custom Attribute:**
- `HBA_Config` - Required for configuration operations.

**Configure NIC Teaming for Windows**

Creates a NIC team on target servers with Windows 2012 or later. The step takes a comma separated list of parameters which specify exactly which NICs will be teamed together.

**Type:** Windows .BAT

**Required Parameters:**
- `Field Name` - lists the NIC characteristic that will be specified in the other parameters. Valid field names are:
  - Name
  - InterfaceDescription
  - ifIndex
  - Status
  - MacAddress
  - LinkSpeed
- `Values` - The value(s) to be checked for that corresponds to the specified Field Name parameter. Note that no extra spaces should be used when specifying parameters.

Table 8 is an example that represents the NICs on a target server with Table 9 as examples of parameters to provide to the script.

**Table 8: Configure NIC Teaming for Windows sample network adapters on a target server**

<table>
<thead>
<tr>
<th>Name</th>
<th>InterfaceDescription</th>
<th>ifIndex</th>
<th>Status</th>
<th>MacAddress</th>
<th>LinkSpeed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
<td>HP NC553i Dual Port FlexFabric 10G</td>
<td>16</td>
<td>Up</td>
<td>18-A9-05-C5-E1-B7</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>Ethernet 2</td>
<td>HP NC553i Dual Port FlexFabric 10G...#1</td>
<td>15</td>
<td>Disconnected</td>
<td>18-A9-05-C5-E1-B3</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>Ethernet 3</td>
<td>HP NC553i Dual Port FlexFabric 10G...#2</td>
<td>14</td>
<td>Disconnected</td>
<td>18-A9-05-C5-E1-B4</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Table 9: Configure NIC Teaming for Windows sample parameters used for target server with network adapters setup as provided in Table 8.**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status,Up</td>
<td>Teams all NICs that are active</td>
</tr>
<tr>
<td>Name,Ethernet,Ethernet 2</td>
<td>Teams the first and second NIC</td>
</tr>
<tr>
<td>MacAddress,18-A9-05-C5-E1-B3,18-A9-05-C5-E1-B4</td>
<td>Teams the second and third NICs</td>
</tr>
</tbody>
</table>

**Custom Attributes:** None
**Continue SuSE AutoYaST Installation**

Causes the SuSE AutoYaST installation to continue after it was stalled for HP SA Agent injection or other tasks before the reboot into the second phase of the SuSE AutoYaST installation process.

*Type: OGFS*

*Parameters: None*

*Custom Attributes: None*

**Copy Boot Media**

Copies files needed to boot the installer environment to the stub partition.

This is one of the scripts used to prepare a server for a Linux installation. It should follow the *Create Stub Partition* script. If no parameters are given, the script will check the installation media and figure out the files that the installer needs in order to boot.

*Type: Unix*

*Optional Parameters:*

- `base_path_or_img` - A directory in the installation media, or a mountable image file. All other files are relative to this location.
- `"file1" ["file2" ["file3" ...]]` - list of files to copy onto the boot partition.

*Custom Attributes None*

**Create Stub Partition**

Creates a partition on the local disk to load the Linux build image.

This is one of the scripts used to prepare a server for a Linux installation. By default, the boot disk is identified as the first disk in `/proc/partitions`. A single bootable 1 GB primary partition is created with an EXT3 filesystem. The partition is mounted under `/mnt/<device_name>`, using the device's `/dev/` path but replacing `/dev/` with `/mnt/` for the mountpoint. For example, this would be `"/mnt/cciss/c0d0p1"` on an HP ProLiant system. A symlink is created under `/mnt/root` so other scripts can alter the root filesystem if needed.

*Type: Unix*

*Requirements:*

- Existing filesystems on the boot disk, if mounted, must be unmountable. This means interactive sessions cannot have the present working directory beneath the mount point, for example. Any partitions on the boot disk that are currently mounted will be unmounted.

- Target server must be running the Linux OGFS Agent. This script uses the following system utilities: mount, umount, sfdisk, and mkfs.ext3

*Parameters: None*

*Custom Attributes None*

**Create Windows System Drive**

Creates a temporary partition on the target server hard drive using diskpart to prepare for Windows operating system deployment. The partition is created within this script versus partitioning within the unattend file for the following reasons:

- HP felt it was easier to create or extend custom partitions versus inside the unattend XML format.
- An error from diskpart may be more intuitive, as well as the build plan will fail at this step without the need to call Windows Setup and look for the Windows installation logs for an error.
- The HP-provided OSBPs use this temporary partition to store the extracted driver files. Although it may be possible to store the driver files on the WinPE X: drive, there is possibility to run out of storage.

If additional partitions are needed, please consider the following options a guideline:
• Option #1
Make a copy of the Create Windows System Drive script and modify it to create the additional partitions using the diskpart command. With this method, all the partitioning is still done within the script.

• Option #2
Modify an HP-provided Windows OSBP
1. Remove the Create Windows System Drive step from the Windows OSBP.
2. Add the boot disk partitioning to the Windows unattend file following the instructions provided by Microsoft. Also, modify the <DriverPaths> section to use X:\$oem$. The latter is necessary since the C: partition will no longer be available with the removal of Create Windows System Drive step.
3. Remove the HP-provided unattend file from the Windows OSBP and replace with the new unattend file that contains the disk partitioning.
4. Update the ProLiant Drivers for Windows XXX OSBP step to use X:\$oem$ as the directory path parameter. This is necessary since C: partition will no longer be available with the removal of Create Windows System Drive step.

NOTE: The X: (WinPE) drive is a RAM disk that only exists in memory. It is possible that the combined size of the drivers may be too large to fit in the available RAM disk space. If this occurs, please consider using Option #1 above.

Type: Windows .BAT
Parameters: None
Optional Custom Attributes:
- SystemDrive
- SystemDiskNumber

Decommission Server
Decommissions the secure agent connection between a target server in production and the appliance to allow for reprovisioning of the target server.
Target servers provisioned by IC server provisioning contain certificates that uniquely and securely identify the server to the appliance. The appliance expects to retrieve this information every time the target server boots. This can cause problems when attempting to reprovision a server or when a server’s hard drive is erased, as the service OS may not be able to retrieve these certificates. This script decommissions a server, breaking that security bond, and allowing a server to boot into maintenance without requiring the identification certificates.
If this script is run against a server that does not have an established secure agent connection, the script exits and the build plan continues without error.

CAUTION: Because of the way this script works, it can only be placed in a build plan after the Boot step and before the Wait for SA Agent step. Placing this script anywhere else will likely cause the build plan to fail.

CAUTION: This script breaks the secure agent connection between the target server and the appliance. Only run this script against a target server if you are certain you will never boot the server back into its production operating system.

Type: OGFS
Parameters: None
Custom Attributes: None

Delete iLO User
Deletes an iLO user using the ProLiant Scripting Toolkit hponcfg utility.

Type: OGFS

Required Parameters:
• --username=USERNAME - Username of the iLO user to delete.
**Deploy Agent**

Deploy an SA agent to the target server.

This script is typically used to deploy an agent to the stub partition in preparation for a Linux installation. This agent is used to communicate with the server during the operating system installation and is not the same agent that is placed on the production server.

**Type:** Unix

**Required Parameters:**
- `-d DEST, --dest=DEST` Where to download the agent. This can be both a file or a directory, absolute or relative. Directories will be created if they don’t exist. The default is to create a file in the current directory with the same name. Directories that don’t exist will be created, note however that the basename will be considered as a file.

**Optional Parameters:**
- `-n NAME, --name=NAME` The name of the file to download.
- `-g SERVER, --gateway=SERVER` Optional server to download from. If not present the agent config will be used to determine a valid gateway.

**Custom Attributes:** None

**Embed files initrd**

Embeds a list of files or directories into a new initrd image.

Adds the specified files to the initrd used for installing the operating system. This is a way of making utilities and drivers available to the Linux installation.

**Type:** Unix

**Required Parameters:**
- `-s NAME` The name of the file to embed. Multiple files can be specified, but each needs its own `-s` parameter.
  
  Ex: `-s file1 -s file2 -s file3`

**Custom Attributes:** None

**Erase Server Disk**

Erases the partition table on all detected disk drives.

This script is used to clear out a server’s disk drives, usually in preparation for reprovisioning.

**CAUTION:** Running this script causes data loss on all connected disk drives.

**Type:** Unix

**Parameters:** None

**Custom Attributes:** None

**Find SD Card on Server**

This script searches for the special embedded SD card device on ProLiant target servers. If a card is found, the custom attribute, `boot_disk`, is set to the device associated with the card. The `boot_disk` custom attribute is used in the Create Stub Partition script step to force the creation of the initial installed OS boot partition on a specific disk.

**Type:** Python

**Parameters:** None

**Custom Attribute:** `boot_disk`
Get Deployment Interface Details
Retrieves and stores Deployment Interface Network details into the specified file. This is used with Install Windows SPP script to reset the network after a NIC firmware or driver update.

Type: Python
Required Parameter:
  - absolute_filename – The absolute path to the file to write the network details.

Custom Attributes: None

Inject AutoYaST Personalization Settings
Inject Kickstart Personalization Settings
Inject Kickstart Personalization Settings for ESXi 5
Injects settings for personalization into the appropriate kickstart or AutoYaST answer file.

When you run a build plan and select the Configure static network information option, a custom attribute named hpsa_netconfig is automatically created for each target server the build plan is run against. hpsa_netconfig contains the network settings to be applied to that particular target server.

At run time, these scripts edit the Kickstart or AutoYast files and insert static IP addressing information based on the contents of the hpsa_netconfig custom attribute. If hpsa_netconfig is not defined, then this script does nothing and completes successfully.

Type: OGFS
Parameters: None
Optional Custom Attribute: hpsa_netconfig

Inject Personalization Settings
Injects settings for personalization into a Windows unattend answer file.

When you run a build plan and select the Configure static network information option, a custom attribute named hpsa_netconfig is automatically created for each target server the build plan is run against. hpsa_netconfig contains the network settings to be applied to that particular target server.

At run time, this script edits the unattend file and inserts the static IP addressing information based on the contents of the hpsa_netconfig custom attribute. If hpsa_netconfig is not defined, then this script does nothing and completes successfully.

Type: OGFS
Optional Parameter:
  - PATH_TO_FILE where an alternate path to the Unattend.xml, Unattend.txt, or sysprep.inf file may be specified. This parameter can normally be omitted and the script will use the standard file locations. The following path locations are checked (in order), and if that file exists, it is updated and no further files are checked.
    X/Windows/Temp/Unattend.xml (For Windows 2008 OS Media installs), C/Windows/Panther/unattend.xml (For Windows 2008 WIM installs).

Optional Custom Attribute: hpsa_netconfig

Inject Required AutoYaST Settings
This script injects required settings into the autoinst.xml file.
This script modifies the AutoYaST file and adds scripting that will allow IC server provisioning to monitor and control the operating system installation.
  - required pre- chroot- and post- exitpoint scripts.
  - pre-exitpoint to start the agent before the installation starts.
• chroot-exitpoint to stall the installation before the AutoYaST reboot between
  o the first and second installation phase. This also triggers
  o the integration of the HP SA agent installation executed
  o during the next reboot.
• post-exitpoint to write info file that the installation has finished

Type: OGFS

Required Parameter:
  • HP SA Agent Name - The SA agent that needs to be started during installation.

Custom Attributes: None

Inject Required ESXi 5 Kickstart Settings
This script injects required settings into the kickstart answer file.
This script modifies the Kickstart file and adds scripting that will allow IC server provisioning to monitor and control the operating system installation.

Type: OGFS

Optional Parameter:
  • accept-encrypted-password - accept the password to be encrypted, which for ESXi is not typically allowed.

Custom Attributes: None

Inject Required Kickstart Settings
This script injects required settings into the kickstart answer file.
It appends code to the %pre section of the kickstart file, to start the agent before the installation starts. It also injects code into the %post section to prevent the installer from rebooting after the installation is complete. The later gives the target a chance to integrate the HP SA agent.

Type: OGFS

Parameters: None

Custom Attributes: None

Inject Required Unattend.xml Settings
This script injects required settings into the unattend answer file.
This script modifies the unattend file and adds scripting that will allow IC server provisioning to better monitor and control the operating system installation.

Type: OGFS

Optional Parameter:
  • PATH_TO_FILE - An alternate path to the Unattend.xml file may be specified. This parameter can normally be omitted and the script will use the standard file locations. The following path locations are checked (in order), and if that file exists, it is updated and no further files are checked: X/Windows/Temp/Unattend.xml,
    C/Windows/Panther/unattend.xml

Custom Attributes: None

Inject Windows Domain or Workgroup Personalization Settings
This script injects Windows Active Directory Domain or Workgroup related configuration into the unattend answer file as part of a Windows OSBP. Values are read from custom attributes and inserted into the unattend file. Supported for Windows Server 2008 and later releases.
Type: OGFS

Optional Parameter:

- `path_to_file` - the OGFS path or absolute path on the target server to the unattend answer file. If this parameter is omitted, the script will look for an existing file at `\Windows\Temp\Unattend.txt`.

Required Custom Attributes:

- `DomainName` – Fully Qualified Domain Name
- `DomainUser` – domain user with permissions to join the domain
- `DomainPassword` – plain-text domain password to join the domain

Optional Custom Attribute:

- `Workgroup` – workgroup name if joining workgroup and not a domain

Install bootloader for ESXi
Installs the ESXi boot loader.

The high level steps are: (1) Writing the EXTLINUX Master Boot Record to the boot disk, (2) Installing the EXTLINUX boot loader, and (3) Creating the extlinux.conf configuration file. This script does not reboot the target server, but when this script completes, the target server may be rebooted and the ESXi scripted install will occur. This script comes after the Create Stub Partition and Copy Boot Media script steps, and expects the following files in /tmp: extlinux, mbr.bin, and ks.cfg.

Type: Unix

Optional Parameter:

- `--kernel_arguments='args'` - Pass on kernel arguments

Custom Attributes: None

Install bootloader for RedHat Enterprise Linux Server
Install bootloader for SuSE Linux Enterprise Server

Installs GRuB (GRand Unified Boot Loader) onto the stub partition.

GRuB is placed in the stub partition in order to enable the booting into the appropriate Linux installation image. The high level steps are: (1) Installing the GRuB boot loader and (2) Configuring GRuB. This script does not reboot the target server. This script comes after the Create Stub Partition.

Type: Unix

Optional Parameter:

- `kernel_arguments='value'` - a string of arguments that will be passed to the build images’ kernel, for example, `kernel_arguments=mpath` used with Red Hat EL 5.8 for multipath support.

Custom Attributes: None

Install Linux SPP
Install Windows SPP

Installs the HP Service Pack for ProLiant (SPP) on the target server.

These scripts install the entire SPP onto the running production target server using the HP SUM utility and specified SPP version. This includes agents, drivers, and firmware. If you do not want to install all of these things, you can control what gets installed using the `--hpsum_parameters` parameter. See the HP SUM User Guide for information about what options are available.

This script expects the SPP location on the Media Server to be \Media\spp\yyyy.xx where `yyyy.xx` is the SPP version number, for example 2013.02. This script requires that the SPP be available, so it should come after the Set
Media Source script. Since this script uses the HP SUM utility, an SPP should be used in the media server location. Copying a supplemental package to that location will fail the script since it won’t be able to find HP SUM.

The SPP version may have an alpha number after the yyyy.xx, for example, 2013.09B. Adding this alpha number at the end will work appropriately to pickup 2013.09B as the latest compared with 2013.09 or 2013.09A.

NOTE: In some cases, it is common for a small percentage of the SPP components to fail. Because of this, by default, the Install xxx SPP script will not report a failure if individual components fail with HP SUM exit code 253. Unfortunately, there is no way to distinguish between one of these expected failures and an unexpected failure. If you prefer that your script and build plan fail when a component fails, you can specify the –fail_on_warning flag.

Type: Unix or Windows .BAT

Required Parameter:

- filename=absolute_filename – The absolute file path containing the Deployment Interface details of the server which can be obtained by running Get Deployment Interface Details script.

Optional Parameters:

- --spp_version=directory_name - The name of the directory containing the SPP to be installed, such as 2013.02. If the value is blank or latest, the script automatically selects the directory with the latest version as determined by sort order.
- --hpsum_options="option1 option2 option3 ... optionN" - HP SUM utility supported command-line options that are to be passed to HP SUM. Refer to HP SUM’s CLIhelp.txt documentation file for available options.
- --fail_on_warning - When specified, the script will fail on receiving the error code 253 from HP SUM. When absent, the script will be successful for the error code 253 from HP SUM

Custom Attributes: None

Integrate HP SA Agent

Sets up the installation of the production SA agent as part of a Windows installation.

The important features it includes are as follows:

- Downloading the HP SA agent installer
- Creating a script that will execute the HP SA Agent installer when the final OS first boots.
- Copying the server’s unique MID and cryptographic certificates into place
- Integrating with Windows setup to schedule a task that will install the HP SA agent when Windows first boots.

For Windows 2008 and newer, this is done by adding the appropriate script commands to the Unattend.xml file

Type: OGFS
Parameters: None
Custom Attributes: None

Integrate Linux HP SA Agent

Sets up the installation of the production SA agent as part of a Linux installation.

The agent will install and register at the first target server boot.

Type: OGFS
Optional Parameter:

- HP SA Agent Name - The name of the HP SA agent to install. The name of the agent doesn’t require quotes. For example, Red Hat Enterprise Linux Server 5.

Custom Attributes: None

Manage iLO Configuration

Captures or sets the target server’s iLO configuration using the ProLiant Scripting Toolkit hpncfg utility.
Manage Smart Array Configuration
Captures or sets the target server’s Smart Array configuration using the ProLiant Scripting Toolkit HP ACU utility.
This script runs in the Linux service OS and attempts to unmount any partition mounted on the Smart Array. The partitions are left unmounted (i.e. not remounted). These partitions must not be in use at the time this script is run.

Type: Unix

Required Parameters:
- To capture: `-c output_filename`
- To set: `-i input_filename`

Optional Parameters:
- `-nofail` - indicates if the script should not fail if a Smart Array controller is not found.
- `-internal` - limit operation to internal controllers. Is not used with `-external` parameter.
- `-external` - limit operation to external controllers. Is not used with `-internal` parameter.
- `-reset` - removes existing data and overwrites current configuration with new configuration. Use only with `-i` parameter.

Custom Attribute: `configuration_location` – The absolute filename of the captured configuration xml on the target server used with the *Upload Configuration File* script step.

Manage System Configuration
Captures or sets the target server’s BIOS system configuration using the ProLiant Scripting Toolkit conrep utility.

Type: Unix

Required Parameters:
- To capture: `-s output_filename`
- To set: `-l input_filename template_filename`

Custom Attribute: `configuration_location` – The absolute filename of the captured configuration xml on the target server used with the *Upload Configuration File* script step.

Monitor Installation
Monitors an installation by checking the installation log file at specified intervals.
This script checks the specified log file at given intervals to see if it has changed. If the log file has not changed after the maximum number of checks, the script fails and reports a timeout. This means that the script will time out after `number_of_checks` multiplied by the timeout in seconds.

Type: OGFS

Required Parameters:
- `log_file` - the log file that needs to be monitored

Optional Parameters:
- number_of_checks - the number of times to check if the log file has changed before failing. This has a default value of 200 times.
- timeout - the timeout between checks. This has a default value of 6 seconds.

Custom Attributes: None

**Prepare Disks on HP ProLiant Gen8**
Prepares disks of an HP ProLiant Gen8 target server and sets the target device custom attribute, SystemDiskDevice.

The embedded flash drives on ProLiant Gen8 servers are sometimes seen by the operating system installers and are possibly used for the hard drive by the installers. This script attempts to hide those embedded flash drives from the operating system installer, and also sets a custom attribute to indicate to the installer, exactly which drive to install. This script is required for Windows operating system deployments since other iLO devices are displayed when the target server is booted to the Intelligent Provisioning WinPE service OS.

Type: OGFS
Parameters: None
Custom Attributes: SystemDiskDevice

**Prepare Windows for Image Capture**
This script prepares a running Windows installation (Windows 2008 and newer) from a Managed server to be captured using the Sysprep tool.

For more information about the Sysprep tool, refer to the following articles:

- [http://technet.microsoft.com/library/cc766514](http://technet.microsoft.com/library/cc766514)

Type: Windows .BAT
Optional Parameter:
- /unattend:path_to_answer_file – Path to the answer file that is passed to the Sysprep tool.

Custom Attributes: None

**Prevent WIM File Overwrite**
Checks for the existence of the WIM image file.

This ensures that an existing WIM image is not accidentally overwritten if another capture build plan is run that uses the same WIM file name. The script returns an error if the specified file name exists on the Media Server. Removing this step from a build plan will allow an existing image file to be overwritten.

Type: Windows .BAT
Required Parameters:
- WIM_filename    The WIM file name and path to be checked if exists.

Custom Attributes: None

**Reboot**
This script reboots a target server into the production operating system.

This script clears any previous boot configuration for the target server and triggers a reboot so that the target server will follow its normal boot order and boot from its hard disk into the installed production operating system.

Type: OGFS
Parameters: None
Custom Attributes: None
**Reboot to Apply BIOS Changes and Power Off**

Implements the BIOS reset by booting the target server and then powering it off.

When a system BIOS is reset to factory settings, the reset does not take effect until the next time the target server is power cycled. This script performs that power cycle, waits a specified amount of time for the reset to take effect, and then powers the server off again. The default settings will then have been applied, and the server is ready to have another build plan run against it.

This script should always be used following the *Reset System Configuration* script step.

**NOTE:** After completion of this script, the target server is left powered off. The next build plan run against this server needs to have a `Boot` step up front to bring the server back up.

**NOTE:** This script is based on the `Boot` script, so it uses some of the same parameters, even though the target server should never actually complete the booting process. It should be powered off before it completes the power on self-test.

**Type:** OGFS

**Required Parameters:**

- `--serviceOS='SERVICE.OS'` - represents the service OS into which to boot; See the method parameter for the supported service OSs. The target server should power off before booting to this OS, so what you put here does not matter much.
- `--force` - Forces the boot configuration and reboot of the target server. This is required to force the boot operation to take place.

**Optional Parameters:**

- `--method='METHOD'` - This parameter is not required as the target server should power off before the boot ever begins, but it can be specified without consequences. Possible values:
  - `auto` - it will behave as embedded or network, depending on the target server
  - `embedded` - supported only if the IloService is enabled on the target server and this is a HP ProLiant Gen8. The following service OSs are possible: `linux, linux64, or winpe64`.
  - `network` - configures PXE network booting into the selected Service OS if the IloService is enabled on the target server it will also set the one time boot option to `NETWORK`. Specifying an ogfs PXE-option as the Service OS is supported.
- `--wait <time>` - Specify a new wait time. Default is 45 seconds.

**Custom Attributes:** None

---

**Remap Windows Drives**

Remaps the volumes to new drive letters, starting with “C”. Network drives are not remapped since those drive letters are explicitly assigned. The "X" drive is reserved for WinPE and is not remapped either.

**Type:** Python

**Optional Parameter:**

- `--reservedDriveLetters="letter1 letter2 ... letterN"` - A space separated list of drive letters that are not to be assigned during the remapping.

**Custom Attributes:** None

---

**Reset System Configuration**

Resets the target server's BIOS system configuration to factory default settings using the rbsureset utility.

As part of the reset, the date and time on the target server is reset.

**NOTE:** The system settings are not actually reset until the next server power cycle. Always follow this step with the *Reboot to Apply BIOS Changes and Power Off* step which will perform this power cycle.
**Run Windows 2008 R2 x64 Setup**
**Run Windows 2008 x64 Setup**
**Run Windows 2012 x64 Setup**

Starts the appropriate Windows operating system installation by call the setup.exe specified in the script parameter.

Type: Windows .BAT

**Required Parameters:**
- “setup.exe_path_and_file” - Path and filename to the Windows setup.exe program. Parameter should include the double quotes.

Custom Attributes: None

---

**Set Media Source**

Defines the connection from the target server to the Media Server.

The **Set Media Source** script is where the location of the media to be used for a build plan is specified. The script will do different things depending on the protocol specified in the URI:

- **SMB** – This is the Samba/Windows file share protocol. When this protocol is specified, the Windows file share is mounted immediately as part of this script.

- **HTTP** – When the HTTP protocol is specified, this script writes the media location information to a special place where it can be read by the Inject Required Settings steps later on. Those steps modify the OS installation answer file to use the HTTP location specified for the OS installation.

- **NFS** – When NFS is specified, the NFS mount happens immediately as part of the script.

**Note:** The media server cannot be a domain controller.

Type: Python

**Required Parameters:**
- **URI** - where URI can be any of the supported schemes:
  - smb from a Windows client:
    ```
    smb://username:password@server/path/to/media[#X:]
    
    where X is the drive letter to mount. If the mount drive is omitted, the default is Z: and the username can include a domain such as domain\username and can be a maximum length of 20 characters.
    ```
  
  - smb from a Linux client:
    ```
    smb://username:password@server/path/to/media#/mount/point?noserverino
    
    where /mount/point is the local mount point and noserverino is required and the username can include a domain such as domain\username and can be a maximum length of 20 characters.
    ```
  
  - http:
    ```
    http://server/path/to/media
    ```
  
  - nfs:
    ```
    nfs://server/path/to/media[#/path/to/mount/point]
    
    If the mount point is omitted, it defaults to /mnt/media
    ```

Custom Attributes: None
**Set One Time PXE Boot**
This script will set one-time PXE boot. It is intended for servers migrated from IC server deployment/RDP. It uses hponcfg to set the one-time PXE boot on Windows-based target servers and /sbin/hpbootcfg on Linux-based target servers. /sbin/hpbootcfg is a utility installed on RDP-deployed target servers.

NOTE: This script will only work on target servers that were provisioned using RDP and migrated to IC server provisioning.

Type: Unix
Parameters: None
Custom Attributes: None

**Uninstall HP Agents and Utilities**
Uninstalls HP agents and utilities that store system specific information in the registry.

This script is typically used with image capture build plans. There may be problems if a Windows image captured with these agents and utilities installed is deployed to another server.

Type: Windows VMScript
Parameters: None
Custom Attributes: None

**Unmap Network Drive**
Unmaps the network drive that is associated with the specified drive letter. If there is no network drive associated with the specified drive letter, this script still completes successfully.

Type: Python
Required Parameters:
- `--driveLetter` - The letter that's associated with the network drive to be unmapped.

Custom Attributes: None

**Unmount All Boot Disk Partitions**
Unmounts all the partitions belonging to the boot disk. By default, the boot disk is computed as the first hard drive listed in /proc/partitions.

Type: Unix
Parameters: None
Optional Custom Attributes:
- `boot_disk` - The absolute path to the device file for the boot disk. For example, /dev/sda.

**Unmount Intelligent Provisioning WinPE Volume**
Unmounts the Intelligent Provisioning WinPE volume containing the directory, $WinPEDrivers$ where WinPE drivers are provided for ProLiant Gen8 servers. This prevents the Windows setup utility from using these drivers for the Windows installation. This script is currently only used with Windows 2008 SP2 deployments.

Type: Windows .BAT
Parameters: None
Custom Attributes: None
**Update Firmware Using SPP**

Runs an off-line firmware update on the target server using HP SUM from an SPP on the media server.

This script must be run while booted in the LinuxPE service OS. It uses the HP SUM utility and specified HP Service Pack for ProLiant (SPP) version. This script expects the SPP location on the Media Server to be `\Media\spp\yyyy.xx` where `yyyy.xx` is the SPP version number, for example 2013.02. This script comes after the Set Media Source script.

Since this script uses the HP SUM utility, an SPP should be used in the media server location. Copying a supplemental package to that location will fail the script since it won't be able to find HP SUM.

The SPP version may have an alpha number after the `yyyy.xx`, for example, 2013.09B. Adding this alpha number at the end will work appropriately to pickup 2013.09B as the latest compared with 2013.09 or 2013.09A.

Since this is an off-line firmware update, the log files showing what happened will be lost as soon as the server reboots. If you want to save a copy of the logs on the media server, use the `hpsum_logs_dump_dir` option as described below.

Type: Unix

Optional Parameters:

- `--spp_version=directory_name` - The name of the directory containing the SPP to be installed, such as 2013.02. If the value is blank or `latest`, the script automatically selects the directory with the latest version as determined by sort order.

- `--hpsum_options="option1 option2 option3 ... optionN"` - HP SUM utility supported command-line options that are to be passed to HP SUM. Refer to HP SUM's CLIhelp.txt documentation file for available options.

- `--hpsum_logs_dump_dir=writable_directory_under_mounted_file_share` - Copies the zipped HP SUM log files to the specified directory. The directory must start with the mount point specified in the Set Media Source script step. For example, if the file share where the zipped HP SUM log file is to be copied to is mounted on `/mnt/ms` and the destination directory is "hpsum_logs", then specify
  
  `--hpsum_logs_dump_dir=/mnt/ms/hpsum_logs`

- `--no_show_log` - Do not display the `hpsum_log.txt` contents in the job log.

Custom Attributes: None

**Validate Custom Attributes**

Validates that the specified custom attributes exist and have values assigned.

This script is used early in a build plan to check that required custom attributes are defined before proceeding with the rest of the build plan. It validates the custom attributes defined in the parameter field and fails the build plan if any custom attribute does not have a value associated with it. Indicating no parameters will cause this script to do nothing and end successfully.

Type: OGFS

Required Parameters:

- `--custAttrNames "custAttrName1 custAttrName2 custAttrName3 ... custAttrNameN"` - The parameter list is a space-separated list of custom attribute names contained inside double quotes.

Custom Attributes: None

**Validate Gateway Setting for Static Network Configuration**

Verifies that a gateway (gw) has been specified when using static network configuration. Some operating systems require a gateway when doing a static network configuration.

When a build plan is run and the **Configure static network information** option is selected, a custom attribute named `hpsa_netconfig` is automatically created for each target server on which the build plan is run. `hpsa_netconfig` contains the static network settings that are applied to that particular target server. This script is used early in the build plan to check if a gateway is defined in `hpsa_netconfig` before proceeding with the rest of the steps in the build plan. The build plan will fail if the gateway is not specified. If `hpsa_netconfig` does not exist, this script will skip the gateway validation and proceed with the remaining steps in the build plan.

Type: OGFS
Required Parameters: None
Optional Custom Attribute: hpsa_netconfig

**Validate ImageX Package Contents**
Verifies that the ImageX package contains `imagex.exe`.

This script is used as a validation step to verify that the imagex utility has been properly uploaded to the appliance. It is used so that the build plan can check for this, warn the user, and fail early in the build plan, before irreversible changes are made to the target server. This step should always follow the deploy imagex package step.

By default, the ImageX package does not contain the imagex.exe utility. Using the IC server provisioning WinPE Image Generation utility, which is available for download from the Settings > DHCP page, a zip package containing WinPE, as well as the required imagex.exe utility is created. When this zip package is uploaded to the appliance in the Settings > DHCP page, using the WinPE Upload button, the imagex.exe utility will be automatically inserted into the ImageX package.

This script validates that imagex.exe made it into the ImageX package and has been installed in the `SystemDrive\HPPROVTEMP` directory using the Install ZIP Package - ImageX step if the target server is in the production OS, or in `X:\Windows\System32` if the target server is in WinPE service OS where SystemDrive is the Windows production system drive letter. On the production OS, the `SystemDrive\HPPROVTEMP` directory is removed after the contents of the ImageX package have been verified.

Type: Windows .BAT
Parameters: None
Custom Attributes: None

**Wait for ESXi installation**
Waits for an ESXi installation to complete.

Since there is no SA Agent available under ESXi, this script is used to verify that the production hypervisor is booted properly. In the ESXi kickstart file, a few lines of Python script executes under first production boot, sending a ping back to say that installation is complete and the machine is booted successfully in to production.

Type: OGFS
Parameters: None
Custom Attributes: None

**Wait for HP SA Agent**
Waits for the HP SA Agent to register or come online before proceeding with future instructions.

This script is used in nearly every build plan. It holds up a build plan’s execution during the time a target server is booting, and will not allow processing to continue until the target server finishes booting and an agent on that server registers itself with the appliance. Any Boot or Reboot step must be followed by a Wait for SA Agent step, since the Boot steps do not wait for boot completion.

Once an agent does register with the appliance, the script will check to make sure the target server was booted into the proper mode as specified by the parameters listed below.

This step can also be used to re-establish communication with a target server when the running build plan might cause a network interruption. If a build plan step causes the target server’s NIC to go down for any reason, the agent will lose its connection to the appliance and cause the build plan to fail. Adding the Wait for HP SA Agent step after the step with the network interruption will cause the build plan to wait for the target server to reestablish a connection.

NOTE: The Decommission Server step is the only step that is allowed to come between Boot and Wait for SA Agent.

Type: OGFS

Optional parameters:

- `--ogfs, --maintenance` Interchangeable parameters that indicate the server is to boot into a service OS. HP-provided build plans do not use `--ogfs`. 

• --production Indicates we are expecting a production agent.

• --atLeast=MINUTES – where MINUTES is the number of minutes to wait before actively checking for the agent; default value is 1 minute. This is necessary in some cases, because this script could actually end up polling for an agent before the target server shuts down. The delay allows the server to shut down and begin its reboot.

• --atMost=MINUTES – where MINUTES is the maximum number of minutes to wait for the agent to come back online; default value is 15 minutes. It may be necessary to increase this setting if the target server is slow at booting and not providing enough time for the agent to connect.

Configuration Files

Summary
The configuration files listed in this section are used or were created to be used by the HP-provided build plans. There are configuration files that are not part of the HP-provided build plans, but are meant to be interchanged with configuration files in the build plans.

ESXi 5.0 U1 Kickstart
ESXi 5.0 U2 Kickstart
ESXi 5.1 Kickstart
ESXi 5.1 U1 Kickstart
ESXi 5.5 Kickstart

Sample answer file for the specified operating system.

Refer to the appropriate operating system supplier for answer file details.

NOTE: For ESXi 5.1 and later, ESXi naming has changed to vSphere; however, OSBPs, scripts and configuration files may continue to state ESXi.

Optional Custom Attributes:

• root_password
• kernel_arguments
• ProductKey_ESXi50, ProductKey_ESXi51 or ProductKey_ESXi55 depending on which file is used - Requires editing answer file to remove the comment for the vmserialnum line.

iLO Configuration - Set Minimum Password Length
Sets the iLO Minimum Password Length with default as 8.

This is a sample file used as a template for creating custom files. Refer to ProLiant Scripting Toolkit documentation for hponecfg configuration file details.

Custom Attributes: None

ImageX Configuration - Exclude Boot Directory
Excludes the /Boot directory from the capture process using the imagex.exe /capture option.

This is required for some Windows capture build plans to prevent multiple boot loader entries from being created when the WIM image is installed, since bcdboot.exe will create a new boot loader entry.

Custom Attributes: None
Sample answer file for the specified operating system. By default with IC server provisioning 7.2.2 and later releases, firewall settings are disabled within these answer files to allow SA agent communication on port 1002. Commented code to enable firewall is also provided that is designed to allow communication on port 1002.

Refer to the appropriate operating system supplier for answer file details.

Optional Custom Attributes:

- root_password
- kernel_arguments

**Smart Array Configuration - Delete RAID Logical Volumes**
Clears the Smart Array configuration.

IMPORTANT: Using this configuration file causes data loss.

This configuration file deletes all logical volumes and arrays on all Smart Array controllers. Refer to ProLiant Scripting Toolkit documentation for HP ACU configuration file details.

Custom Attributes: None

**Smart Array Configuration - Set RAID 1**
**Smart Array Configuration - Set RAID 5**
Configures the Smart Array with the specified RAID level

Refer to ProLiant Scripting Toolkit documentation for HP ACU configuration file details.

Custom Attributes: None

**System ROM Configuration - Enable Boot from SAN**
**System ROM Configuration – Enable Virtualization**
Enables Boot from SAN on ProLiant Blade Servers or Virtualization BIOS functionality on ProLiant servers.

Refer to ProLiant Scripting Toolkit documentation for conrep configuration file details.

Custom Attributes: None

**Windows 2008 SP2 DataCenter x64 en_us Unattend**
**Windows 2008 SP2 Enterprise x64 en_us Unattend**
**Windows 2008 SP2 Standard x64 en_us Unattend**
**Windows 2008 SP2 Web Server x64 en_us Unattend**
Sample answer file for the specified operating system and edition.

Refer to the appropriate operating system supplier for answer file details.

NOTE: The partitioning of the boot disk is performed in the previous Create Windows System Drive OSBP step. Refer to the Create Windows System Drive script description in the Scripts section within this document for important information about disk partitioning.

Required Custom Attributes:

- ProductKey_Win2008-DC-x64, ProductKey_Win2008-Ent-x64, ProductKey_Win2008-Std-x64, or ProductKey_Win2008-WS-x64

Optional Custom Attributes:
• ComputerName

• AdminPassword – The default setting within the Windows unattend answer file is encrypted. This value must be encrypted if using the answer file unchanged. To encrypt a Windows Administrator password, run the Windows System Image Manager (Windows AIK) and create a temporary answer file with the password encrypted. The password can then be copied from the temporary answer file and used as the value for this Custom Attribute.

• SystemDrive
• SystemDiskNumber

Windows 2008 R2 SP1 DataCenter x64 en_us Unattend
Windows 2008 R2 SP1 Enterprise x64 en_us Unattend
Windows 2008 R2 SP1 Standard x64 en_us Unattend
Windows 2008 R2 SP1 Web Server x64 en_us Unattend

Sample answer file for the specified operating system and edition.

Refer to the appropriate operating system supplier for answer file details.

NOTE: The partitioning of the boot disk is performed in the previous Create Windows System Drive OSBP step. Refer to the Create Windows System Drive script description in the Scripts section within this document for important information about disk partitioning.

Required Custom Attributes:

• ProductKey_Win2008R2-DC-x64, ProductKey_Win2008R2-Ent-x64, ProductKey_Win2008R2-Std-x64, or ProductKey_Win2008R2-W5-x64

Optional Custom Attributes:

• ComputerName

• AdminPassword – The default setting within the Windows unattend answer file is encrypted. This value must be encrypted if using the answer file unchanged. To encrypt a Windows Administrator password, run the Windows System Image Manager (Windows AIK) and create a temporary answer file with the password encrypted. The password can then be copied from the temporary answer file and used as the value for this Custom Attribute.

• SystemDrive
• SystemDiskNumber

Windows 2012 DataCenter x64 en_us Unattend
Windows 2012 Standard x64 en_us Unattend

Sample answer file for the specified operating system and edition.

Refer to the appropriate operating system supplier for answer file details.

NOTE: The partitioning of the boot disk is performed in the previous Create Windows System Drive OSBP step. Refer to the Create Windows System Drive script description in the Scripts section within this document for important information about disk partitioning.

Required Custom Attributes:

• ProductKey_Win2012-DC-x64 or ProductKey_Win2012-Std-x64

Optional Custom Attributes:

• ComputerName

• AdminPassword – The default setting within the Windows unattend answer file is encrypted. This value must be encrypted if using the answer file unchanged. To encrypt a Windows Administrator password, run the Windows System Image Manager (Windows AIK) and create a temporary answer file with the password encrypted. The password can then be copied from the temporary answer file and used as the value for this Custom Attribute.
The packages listed in this section are used or were created to be used by the HP-provided build plans. There may be packages that are not part of the HP-provided build plans, but can be interchanged with packages in the build plans. There may be packages listed in your appliance that are not listed here. These packages are not used or tested by IC server provisioning and will not be discussed here. Additionally, there are no packages for ESXi deployments since those drivers are part of the HP-provided VMWare ESXi operating system distribution.

**ESXi Installation Utilities**
Contains the boot loader and Master Boot Record needed to boot ESXi.

**GRub Boot Loader x86**
Contains the grub boot loader that will land on the stub partition.
Even though the name contains x86, it is still used with x64 Linux deployments.

**Hponcfg for Windows x64 with One Time PXE Boot**
Contains hponcfg and support files to setup the one-time PXE boot.
This package is part of the RDP migration related build plans. It puts hponcfg and support files needed to setup the one-time PXE boot on a target server running a Windows production operating system. The bundled hponcfg.exe only supports x64 bit architectures
This package should only be used on servers deployed using RDP and then migrated to IC server provisioning

**ImageX**
The ImageX Microsoft imaging utility.
By default, this package does not contain the imagex.exe utility, but instead, contains a dummy file. The real ImageX utility is uploaded to the appliance with the WinPE image that gets generated externally and is then uploaded to the appliance. Using the IC server provisioning WinPE Image Generation utility, which is available for download from the Settings > DHCP page, a zip package containing WinPE, as well as the required imagex.exe utility is created. When this zip package is uploaded to the appliance in the Settings > DHCP page, using the WinPE Upload button, the imagex.exe utility will be automatically inserted into this package.

**LinuxPE add-on packages**
Contains additional libraries and utilities required in the LinuxPE PXE service OS.

**LinuxPE HBA add-on packages**
Contains additional libraries and utilities required by the Fibre Channel utilities in the LinuxPE PXE service OS.
Contains Service Pack for ProLiant (SPP) drivers for the specified operating system. Starting with IC server provisioning release 7.2.2, the package names include the SPP version, yyyy.mm. This is to allow multiple SPP driver versions to reside on the appliance.

ProLiant Scripting Toolkit for Linux x64
Contains the ProLiant Scripting Toolkit 64-bit utilities.

Appendix A

New Changes for IC server provisioning Release 7.2.2

Custom Attribute added for several ProLiant HW build plans
A new custom attribute, configuration_location was added to the following ProLiant HW build plans:

- ProLiant HW - iLO Capture Configuration
- ProLiant HW - Smart Array Capture Settings
- ProLiant HW - System ROM Capture Settings

This custom attribute is the absolute filename where the captured configuration file is to be stored and it will contain a default value. The value is considered to be a temporary location where the file will reside until it is consumed by a later step in the build plan. Adding this custom attribute to any existing build plans is not required unless specific control is necessary where these temporary files are stored.

Script step added to ESXi build plans for network gateway validation
A new build plan script step was added to the beginning of all ESXi scripted installation build plans. This Validate Gateway Setting for Static Network Configuration script build plan step verifies that a gateway was specified if static networking information is provided for the target server. Unlike most other operating systems, ESXi requires that a gateway be specified when configuring static IP information during an installation. The only purpose of this script is to do error checking and fail the build plan before the ESXi installation is started. Adding this step to existing ESXi build plans is recommended as an extra safety check, but is not required.

New ProLiant Driver package naming to include SPP version
For all Windows and Linux scripted install build plans, the ProLiant drivers packages have been named to include the SPP version with format as ProLiant Drivers for xxxx – yyyy where xxxx is the operating system name and version and yyyy is the SPP version. This new naming convention will allow storage of multiple SPP versions of the ProLiant driver packages on the appliance. After an upgrade, all HP-provided build plans will use the latest versions of the driver packages. Any customized build plans will continue to use older versions of the driver packages. To use the latest driver packages for customized build plans, a build plan modification will be required to replace the older ProLiant drivers package build plan step with these new ones.

Windows build plans contain comment information regarding disk partitioning in the unattend answer files
Two commented lines have been added to all default Windows unattend answer files regarding the disk partitioning. The commented lines explain that disk partitioning is done within a script build plan step of the build plan and not within the unattend answer file. These are just comments and are not necessary for customized build plans.
**Linux build plans have firewall disabled in default kickstart or autoyast answer files**

The default Linux answer files have the firewall disabled. Commented example lines are provided to enable the firewall, but still allow communication on port 1002. If configuring a firewall on target servers, make sure port 1002 is open for agent communication.

**Modifications to the Erase Server Disk build plan**

A new build plan script step was added to the Erase Server Disk build plan to set the server life-cycle to UNPROVISIONED and the `wait` parameter to the `Reboot to Apply BIOS Changes and Power Off` script build plan step was increased to 180:

```
--serviceOS=linux64 --force -wait=180
```

This is a recommended change to customized build plans.

**For more information**

To read more about Insight Control server provisioning, go to [www.hp.com/go/insightcontrol/docs](http://www.hp.com/go/insightcontrol/docs)

- HP Insight Management Support Matrix
- HP Insight Control Server Provisioning Installation Guide
- HP Insight Control Server Provisioning Administrator Guide
- HP Insight Control Server Provisioning online help
- HP Insight Control Release Notes

**Get connected**

[hp.com/go/getconnected](http://hp.com/go/getconnected)

Current HP driver, support, and security alerts delivered directly to your desktop

© Copyright 2013 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. 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.