N1 Grid Service Provisioning System User's Guide and Release Notes for the OS Provisioning Plug-In 1.0

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–1467
May 2005

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Adobe PostScript

050428@11223

Contents

Tables

Preface

The *N1 Grid Service Provisioning System User's Guide and Release Notes for the OS Provisioning Plug-In 1.0* explains how to provision operating systems through the N1 Grid Service Provisioning System (N1 Grid SPS) product.

Who Should Use This Book

The main audience for the *N1 Grid Service Provisioning System User's Guide and Release Notes for the OS Provisioning Plug-In 1.0* includes system administrators and operators of N1 Grid SPS software who want to be able to maintain the operating systems throughout their enterprise. These users are expected to be familiar with the following:

- The N1 Grid SPS product
- Standard UNIX™ and Windows commands and utilities
- The general concepts and management features available for the operating systems to be provisioned

Before You Read This Book

If you are not already familiar with using the N1 Grid SPS software, read the following books:

- *N1 Grid Service Provisioning System 5.0 System Administration Guide*
- *N1 Grid Service Provisioning System 5.0 Installation Guide*
- *N1 Grid Service Provisioning System 5.0 Operation and Provisioning Guide*

How This Book Is Organized

Chapter 1 provides an overview of the OS provisioning feature.

Chapter 2 includes known issues and late-breaking news about OS provisioning features.

Chapter 3 describes how to set up the environment to more easily support OS provisioning.

Chapter 4 explains how to install and configure the OS provisioning plug-in.

Chapter 5 provides step-by-step instructions on how to provision Solaris systems.

Chapter 6 provides step-by-step instructions on how to provision Linux systems.

Chapter 7 provides step-by-step instructions on how to provision Windows systems.

Chapter 8 describes appropriate target hosts for OS provisioning.

Appendix A describes the specific variables that apply to each target host type.

Appendix B describes the specific variables that apply to the Solaris provisioning profiles.

Appendix C explains how to extend the Solaris provisioning functionality by creating user-specific custom modules.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

Sun Function	URL	Description
Documentation	`http://www.sun.com/documentation/`	Download PDF and HTML documents, and order printed documents
Support and Training	`http://www.sun.com/supportraining/`	Obtain technical support, download patches, and learn about Sun courses

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. Perform a *patch analysis*. Do *not* save the file. [Note that some emphasized items appear bold online.]

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

CHAPTER 1

Introduction to OS Provisioning

Using the Operating System (OS) provisioning plug-in enables you to provision Solaris, Linux, and Windows operating systems on previously uninstalled systems. The plug-in offers a solution to automate and configure OS provisioning on heterogeneous platforms. The plug-in also provides the ability to track installation history and versions.

This chapter contains the following information:

Overview of OS Provisioning

The OS provisioning plug-in solution reduces the cost and complexity of provisioning an operating system on "bare metal" hardware. This solution automates the manual error-prone processes involved in installing an operating system, without requiring any relocation of the hardware. The plug-in solution not only automates provisioning at a single chassis or system level, but also automates the provisioning of your data centers. This solution provides greater deployment flexibility for system administrators and end users. The solution enables you to automate routine system administration tasks that use off-the-shelf software. The provisioning process is similar, regardless of the hardware, operating system, or underlying technology that you are deploying.

The OS provisioning plug-in solution is based on the N1 Grid Service Provisioning System (N1 Grid SPS) product. The N1 Grid SPS product provides a set of command-line tools and graphical user interfaces. The provisioning software tools enable you to more easily provision an operating system on bare metal hardware that is distributed across a network. Using this approach, you can view the act of provisioning the operating system as a service just like any other software service that is available over the network. You can invoke the OS provisioning feature through a web browser or through the command line.

The system administrator can automatically provision the required operating system on a server without even having to connect to the server. In most circumstances, the OS provisioning plug-in solution does not require that you type a command at the target system's prompt (for example, OBP) or turn on and off power switches. This automated hands-free installation is invaluable for administrators trying to install several hundred machines. Thus, you can push an operating system onto a host from a centralized location rather than initiating the request from the host.

The OS provisioning plug-in solution not only provides a better infrastructure software provisioning solution but also addresses a broader scope of provisioning issues, such as tracking installation history, archiving, and version management. The solution addresses both image life cycle management and server life cycle management. This solution provides a common operational flow by abstracting the intricacies of the different operating system installation mechanisms.

Functions

The OS provisioning plug-in provides the following functions through the N1 Grid SPS browser and command-line interfaces:

- Creating, managing, and registering information for customized OS profiles
- Creating, managing, and registering information for provisionable hosts
- Provisioning operating systems onto provisionable hosts
- Creating, managing, and registering information for boot and install servers
- Creating, managing, and registering information for OS provisioning server

Provisioning Process Overview

Operating system (OS) provisioning is the act of installing a given operating system across several hosts. This process consists of several high-level steps:

1. Configure your hardware to support OS provisioning. A more detailed discussion of appropriate configuration recommendations and requirements is in "Supported Systems" on page 20 and Chapter 3.

2. Install the OS provisioning plug-in, as described in "Installing the OS Provisioning Plug-In" on page 47.
3. Create the OS provisioning server, as explained in "Creating the OS Provisioning Server" on page 48.
4. Define appropriate target hosts to be provisioned. See Chapter 8.
5. Provision the appropriate operating system onto the target hosts. See Chapter 5, Chapter 6, and Chapter 7.

OS Provisioning System Software Architecture

The OS provisioning plug-in provides a framework to provision heterogeneous operating systems onto various target hosts that support different protocols, such as IPMI, ALOM, LOM, ILO, SC, and telnet.

Software Architecture Big Picture

The following diagram provides the functional components in use for the OS provisioning plug-in. The diagram assumes that the N1 Grid SPS Master Server is outside the firewall, and the other components are inside the firewall. The OS provisioning server assumes that the user set up the network layer 2 connectivity, which consists of subnets, VLAN's, and so on. All the network information is registered with the OS provisioning server.

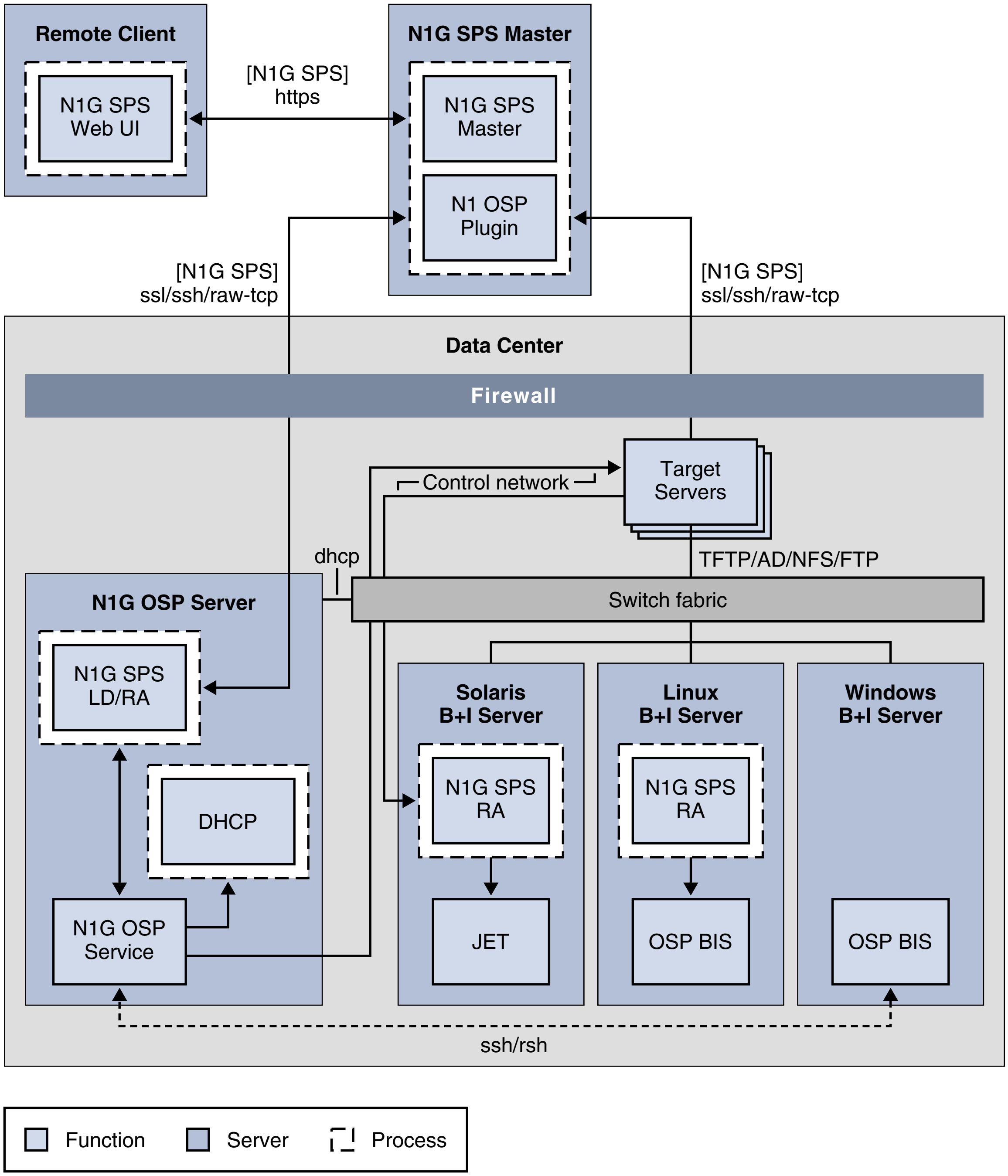

FIGURE 1–1 Software Architecture Diagram

The figure describes the following relationships among the OS provisioning components and uses slightly abbreviated terminology:

- Remote Client – The N1 Grid SPS remote client runs the browser interface and command-line interface. The remote client can be a separate system from the Master Server.
- N1G SPS Master – The N1 Grid SPS Master Server is the main processing engine of the N1 Grid SPS software.
- N1 OSP Plug-In – The OS provisioning plug-in is installed on the Master Server. The plug-in provides functionality to install operating systems on different hardware platforms that support different protocols.
- N1 OSP Server – The OS provisioning control server, usually referred to as the OS provisioning server, is the main processing engine of the OS provisioning plug-in. The OS provisioning server runs the OS provisioning service (N1G OSP Service), which orchestrates the OS provisioning activities. The OS provisioning server controls the target hosts through a control network using appropriate network management protocols (such as IPMI, ALOM, LOM, ILO, and terminal server). These protocols over the control network are used to automate the power, boot, and console services.

  The OS provisioning server supports extensive network topologies (multiple subnets, VLANs, and so on). The OS provisioning server has a bundled DHCP server to serve relevant IP addresses and other boot specific information to target hosts .
- Boot and Install Servers – Three servers are shown supporting OS specific Boot and Install servers:
  - Solaris B + I Server – The Solaris boot and install server uses the JumpStart™ Enterprise Toolkit (JET) to automate the installation of the Solaris distribution media and installation profile.
  - Linux B + I Server – The Linux boot and install server uses the Linux Kickstart technology.
  - Windows B + I Server – The Windows boot and install server uses Windows Remote Installation Services (RIS) technology.

  The boot and install servers have OS-specific boot and install services for automation and monitoring purposes. You have to set up the Linux and Windows boot and install servers outside of the OS provisioning plug-in. For Linux systems, you have to install the N1 Grid SPS Remote Agent (RA) manually. For Solaris systems, the OS provisioning plug-in installs and configures the RA.

Network Protocols

The OS provisioning plug-in uses the following network protocols:

- DHCP – DHCP is used for allocating IP addresses and boot metadata between the OS provisioning server and the target hosts through a switch fabric.

- Network management protocols – Protocols, such as IPMI, ALOM, LOM, ILO and terminal server, are used through a Control Network from the OS provisioning server to the target hosts.
- TFTP, Active Directory, NFS, and FTP protocols – The boot and install servers use the appropriate protocols to communicate and to install the OS onto the target hosts.

Supported Systems

The OS Provisioning plug-in solution provides support for a matrix of operating systems and hardware platforms. This support falls into several categories:

- System components
- OS provisioning components
- Boot and install servers
- Target host hardware platforms
- Target host operating systems

System Components

The following table lists the system components used for OS provisioning.

TABLE 1–1 OS Provisioning System Components

Server	N1 Grid SPS Component	OS Provisioning Component
Master Server	N1 Grid SPS master, agent, and CLI	OS provisioning plug-in
OS provisioning server	N1 Grid SPS RA and CLI	OS provisioning service
Solaris JET server	N1 Grid SPS RA and CLI	OS provisioning boot and install service (BIS)
Linux Kickstart server	N1 Grid SPS RA	OS provisioning boot and install service (BIS)
Windows RIS server	None	OS provisioning boot and install service (BIS)

OS Provisioning Components

The following table lists the provisioning components and their relationships to operating systems.

TABLE 1–2 OS Provisioning Components by Operating System

OS Provisioning Component	Host Operating System
OS provisioning plug-in	Solaris 9, Solaris 10, RedHat AS 3.0, Windows Server 2000
OS provisioning service	Solaris 9 (SPARC)
Solaris boot and install server	Solaris 9, Solaris 10
Linux boot and install server	RedHat AS 3.0
Windows boot and install server	Windows Server 2003

Boot and Install Servers

The following table lists the provisioning technologies that apply to each operating system.

TABLE 1–3 OS Provisioning Technology by Operating System

Operating System	Provisioning Technology
Solaris 9 and 10	Custom JumpStart through JET
RedHat Linux AS 3.0	Kickstart
Windows 2003	Remote Installation Services (RIS)

Target Host Hardware Platforms

Supported target hosts are distinguished by the protocol that the host supports. For a detailed list of supported platforms and associated target host types, see "Target Host Types" on page 95.

Target Host Operating Systems

The OS provisioning plug-in can provision the following operating systems:

- Solaris 9 and 10 (SPARC and x86)
- RedHat Linux 3.0
- Windows 2000 (with service packs)

CHAPTER 2

Release Notes for OS Provisioning Plug-In

This chapter describes late-breaking news and known issues with the OS provisioning plug-in.

The chapter contains the following information:

- "Installation Issues" on page 23
- "Runtime Issues" on page 23
- "Troubleshooting" on page 26

Installation Issues

There are no known installation issues.

Runtime Issues

The following issues are known to exist when provisioning operating systems.

Solaris: Wrong Encryption of root Password Causes JumpStart Error (6245964)

Description: You see the following messages during installation and the installation becomes interactive:

```
root_password=Clz6pK2b6qw=
    syntax error  line 2 position 15
```

The password variable `sysidcfg_root_password_base_conf` in the Solaris OS profile has a encrypted value. However, the password that you supplied was not a Solaris-encrypted password..

Workaround: Use the Solaris tools to encrypt the password. The appropriate mechanism for a user to create an encrypted password is to create a user with a password. Look in the `/etc/shadow` file for the encrypted password and use it in as a value for the `sysidcfg_root_password_base_conf` variable.

Solaris: RA Installer Uses Value of HostName Instead of Target `sps_ra_host` (6255081)

Description: If the OS profile has the value `install_ra_from_snapshot_spsra="n"`, the N1 Grid SPSRemote Agent (RA) listens on the IP address specified by the host name rather than the IP address specified in the variable `sps_ra_host`. In this case, when the RA starts, the `transport.config` uses the host name of the system rather than the value specified in `sps_ra_host`.

Workaround: Create and use a snapshot rather than the RA installer. Follow these steps:

1. Create an RA snapshot. For example, on a Solaris x86 system on which the N1 Grid SPS 5.0 RA is installed, use these commands:

   ```
   #cd /opt/SUNWn1sps/N1_Grid_Service_Provisioning_System
     #cat > /tmp/exclude
     ./agent/data
     <control-D EOF>
     #tar cvf /tmp/exclude ./agent ./common > /tmp/sps_ra_solaris_x86_5.0.tar
   ```

2. Import the snapshot to the JET server, as explained in "How to Import N1 Grid SPS RA Installers" on page 64.

   Note – The media path should point to a directory reachable from JET Server that contains the tar file that you created in the previous step.

Unable to Change Location of OS Provisioning Scripts on Windows Boot and Install Server (6251010)

Description: You cannot change the OS provisioning script location for a Windows boot and install server once the Windows boot and install server is created.

Workaround: Recreate a new Windows boot and install server that has a different name.

Solaris 10 Installation on Target Host Requires User Response (6245773)

Description: If a time server cannot be reached during installation, the installation becomes interactive. During installation, if the time server was not specified, the `sysidcfg` file uses the "Solaris 10 Jet Server" as the time server. On Solaris 10 systems, the boot and install servers do not start the time services by default.

Workaround: There are two ways to resolve this issue.

- Start the time service on the Solaris 10 JET server manually. Type the following command:

  ```
  # svcadm enable time:stream
  ```

- Specify a valid time server (for example, use the address of the OS provisioning server).

Warnings for DHCP Settings Are Not Reported to the User Interface (6248485)

Description: The provisioning operation fails because the DHCP settings are incorrect. There is no message shown in stdout or stderr.

Workaround: The incorrect settings cause the OS provisioning subnet to be created with wrong values. Look at the `/var/adm/n1osp*` log files on the OS provisioning server for the DHCP error.

JET Attach Fails Because `osp_pkgchk.sh` File Does Not Exist (6257748)

Workaround: Install the jet utilities tar file on the JET server physical host prior to the JET server attach.

In the N1 Grid SPS browser interface, follow these steps:

1. In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.
2. On the OS Provisioning Common Tasks page, click Manage in the JET Solaris Image Servers section.
3. Select the referenced component `/com/sun/n1osp/resource/jet_util.tar`.
4. On the Component Details page, click the Run action next to the default:install procedure.
5. Select the physical target host name on which the JET server is installed.

6. Click Run Plan (includes preflight).

Cannot View OS Installation Log by Host Provision Status in Non-EUC Locale (6255797)

Description: Installation log files are always in related EUC locale regardless of the specified locale for the OS installation. When the remote agent locale is different from this EUC locale, you cannot view the log file correctly through the Status Monitoring page because the locales do not match.

Workaround: Connect to service port or console (if applicable) with proper locale to view the log files directly.

Troubleshooting

General Troubleshooting Guidelines

Problem: Provision plan ran successfully, but provisioning on target failed.

Solution: There could be several problems. The paragraphs below list some of the possible reasons why provisioning might have failed. Use this list to isolate the problem.

1. Look into the provisioning logs by viewing the Host Status information in the N1 Grid SPS browser interface. Run the Host Status plan on the target. Check both stdout and stderr to see the reason for failure.
2. Log in to the OS provisioning server and check for logs in the `/var/run/n1osp/log` folder and also check the console output in the `/var/run/n1osp/console` folder. Also, check the messages in `/var/adm/n1osp*` files. To view more detail in the `/var/adm/n1osp*` files, change the value of `n1.isp.core.debuglevel` property in the `/opt/SUNWn1osp/etc/n1osp.properties` file. For example, `n1.isp.core.debuglevel=25`.
3. Obtain a console to the target and re-provision to see reason for failure.
4. Check for network connectivity between the OS provisioning sever, the boot and install server, and the target host. ping/snoop for packets between the OS provisioning sever and the boot and install server, between the OS provisioning server and the target host, and between the boot and install server and the target host.

Note – If the OS provisioning sever or boot and install server has several IP addresses, use the addresses that are used for provisioning.

5. DHCP packets from target are not reaching the OS provisioning sever. Check if the target has been configured to boot over the network using DHCP. If the target host has several network interfaces, ensure that it is using the interface specified in the host profile to boot and install the operating system. Re-provision the target and check if the `/etc/dhcpd.conf` file on OS provisioning server has entries for the target host. The DHCP is configured to respond to target only for the duration of OS provisioning, so you need to re-provision the target to see if the DHCP has been configured properly.
6. Check if the Solaris, Linux, and Windows boot and install servers have been set up properly. Read the appropriate OS documentation for more details. Verify that the boot and install server is properly configured to share the OS media using NFS (for Solaris and Linux) or CIFS (for Windows). Verify that the IP addresses used in the OS profile and the IP addresses configured on the boot and install server match. Check if the TFTP services are configured to run on boot and install server.
7. Check the OS profile information and Host Profile information for IP addresses, Passwords and other information.
8. If an OS fails to install or gets hung, check if your OS profile has the necessary drivers to boot the targets over the network. See the respective OS documentation for more details.

Problem: Error while creating profiles, creating hosts, or provisioning targets.

Solution: Errors can occur at several points in the provisioning process. Check the following:

1. Verify that the N1 Grid SPS Remote Agents (RAs) are installed correctly on the boot and install servers. Verify that the Master Server can reach the RAs. For more information, see the *N1 Grid Service Provisioning System 5.0 Installation Guide*.
2. Verify that the N1 Grid SPS command-line interface (CLI) is installed on the OS provisioning server and the Solaris boot and install server. Run a simple `cr_cli` command.
3. Check the stdout and stderr of the plan.
4. Verify that valid values are provided for the plan and component variables.

Solving Solaris-Related Problems

Problem: I do not understand the sequence of operations for provisioning the Solaris operating system.

Solution: The sequence for Solaris is as follows:

1. The JET server/Solaris boot and install server is prepared for the target host.
2. The DHCP on OS provisioning sever is setup for the target host.
3. The target host is rebooted to boot over network using DHCP.
4. The target host broadcasts DHCP discover packets.
5. The DHCP server on OS provisioning server sends DHCP offer.
6. The target host broadcasts DHCP request packets.
7. The DHCP server on OS provisioning server sends DHCP ACK.
8. The target uses TFTP protocol to get the boot kernel from the JET server.
9. The target installs the OS by getting files over NFS from the JET server.

Problem: While importing a Solaris image, the plan times out.

Solution: Set the default time out for plans on the Master Server. Follow these steps:

1. Edit the following configuration file:

```
/opt/SUNWn1sps/N1_Grid_Service_Provisioning_System_5.0/server/config/config.properties
```

2. Set the following properties:

   ```
   pe.nonPlanExecNativeTimeout=12000
   pe.defaultPlanTimeout=12000
   ```

3. Restart the Master Server.

```
# cr_server stop
# cr_server start
```

Problem: While importing a Solaris image, the plan fails.

Solution: Follow these steps to analyze the problem:

1. Check the stdout and stderr messages of the plan.
2. Verify that there is enough disk space to hold the media.
3. Check the values for the variables. Make sure that all paths are correct and complete.

Problem: Provision plan fails indicating failure in spsra module

Solution: The N1 Grid SPS RA is installed on the target by using either the snapshot of the N1 Grid SPS RA installed on the JET server or by using an RA distribution. Check the `install_ra_from_snapshot_spsra` value in the OS profile. If the target host and the JET server are of different architecture, then you must set `install_ra_from_snapshot_spsra` to “n” and install the N1 Grid SPS RA on the JET server using the "Jet" component.

Problem: While provisioning Solaris x86, installation becomes interactive.

Solution: Ensure that the console variable `x86_console_base_config` in the OS profile is properly configured. For v20z targets, this should be `ttya`. If the installation fails indicating that the boot partition size is small, this is most likely because another OS was previously installed that uses a different disk label format. Use the `fdisk` utility to repartition the disk.

Solving Linux-Related Problems

Problem: I do not understand the sequence of operations for provisioning the Linux operating system.

Solution: The sequence for Linux is as follows:

1. The Linux boot and install server is prepared for the target host.
2. DHCP on OS provisioning sever is set up for the target host.
3. The target host is rebooted to boot over the network using DHCP.
4. The target host broadcasts DHCP discover packets.
5. The DHCP server on the OS provisioning server sends DHCP offer.
6. The target host broadcasts DHCP request packets.
7. The DHCP server on the OS provisioning server sends DHCP ACK.
8. The target uses TFTP protocol to get the boot kernel from the Linux boot and install server.
9. The target installs the OS by getting files over NFS from the Linux boot and install server.

Problem: Installation starts, but user is prompted that the disk label could not be read.

Solution: This problem most likely indicates that another OS was previously installed that uses a disk label format that Linux did not recognize as the default for the architecture. To force the installer to re-initialize the disk label to the default architecture without prompting the user, add the `--initlabel` option to the clearpart directive of the kickstart configuration file.

Problem: Installation cannot get IP address through DHCP.

Solution: Try the following solutions:

- Set the switch port to portfast in the switch to which this target is connected.
- Check the linksleep in PXE configuration file.
- Change InitialBootTimeout in the kickstart properties file to a larger value.

Problem: Target gets the DHCP packet, but fails to boot.

Solution: Try the following solutions:

- Check if the pxelinux.0 and the initrd support the target platform and have the necessary drivers to boot the target over the network.
- Verify that the OS profile and the OS distribution are set up properly on the Linux boot and install server.
- Verify that NFS and TFTP services are configured properly on the Linux boot and install server.

Problem: You see the following message on the console:

```
VFS: mounted root (ext2) filesystem
```

Solution: The Linux kernel has redirected the console elsewhere. Change the console settings in the PXE configuration file.

Problem: Installation goes into interactive mode.

Solution: Check kickstart file for errors. Verify that the server IP address and paths are correct and complete.

Solving Windows-Related Problems

Problem: I do not understand the sequence for provisioning the Windows operating system.

Solution: The sequence for provisioning Windows is as follows:

1. The Windows boot and install server is prestaged for the target host in the active directory.
2. DHCP on OS provisioning sever is set up for the target host.
3. The target host is rebooted to boot over network using DHCP.
4. The target host broadcasts DHCP discover packets.
5. The DHCP server on OS provisioning server and BINL on the Windows boot and install server sends DHCP offers.
6. The target chooses DHCP offer from OS provisioning server and broadcasts DHCP request packets.
7. The DHCP server on OS provisioning server sends DHCP ACK.
8. The target broadcasts again for DHCP discover (for PXE boot server).
9. The BINL on the Windows boot and install server sends DHCP offer (for PXE).
10. The target uses the next server information in the DHCP packet and does a TFTP to get the boot kernel from the Windows boot and install server.
11. The target goes through text mode installation by getting files over CIFS from the Windows boot and install server.
12. The target reboots.
13. By this time the DHCP server is cleared to not respond to the target host, so the target boots from the disk.
14. The target goes through the GUI mode installation.
15. The target reboots and runs the scripts in GuiRunOnce section of the SIF file.

Issues Related to PXE/DHCP/BINLSVC

Problem: How do I know I have the correct PXE ROM version?

Solution: When the NetPC or client computer ROM-boots, a PXE (LSA) ROM message appears on the screen. You can see which version of the PXE ROM code is displayed during the boot sequence of the client machine. Windows 2000 RIS supports .99c or greater PXE ROMs. You may be required to obtain a newer version of the PXE-based ROM code from your OEM if you are not successful with this existing ROM version.

Problem: How do I know if the client computer has received an IP Address and has contacted the Remote Installation Server?

Solution: When the client computer boots, the PXE Boot ROM begins to load and initialize. The following four-step sequence occurs with most Net PC or PXE ROM-based computers:

Note – The sequence may be different on your computer.

1. The client computer displays the message BootP. This message indicates the client is requesting an IP address from the DHCP server.

   Troubleshooting: If the client does not get past the BootP message, the client is not receiving an IP address. Check the following possibilities:

   - Is the DHCP server available and has the service started? DHCP and RIS servers must be authorized in the Active Directory for their services to start. Check that the service has started and that other non-remote boot-enabled clients are receiving IP addresses on this segment.
   - Can other client computers, such as (non-remote boot-enabled clients, receive an IP address on this network segment?
   - Does the DHCP server have a defined IP address scope and has it been activated? To verify this feature, click Start, point to Programs, point to Administrative Tools, and click DHCP. Alternatively, you can click Start, point to Programs, point to Administrative Tools, and click Event Viewer.
   - Are there any error messages in the event log under the System Log for DHCP?
   - Is a router between the client and the DHCP server not allowing DHCP packets through?
2. When the client receives an IP address from the DHCP server, the message changes to DHCP. This indicates the client successfully leased an IP address and is now waiting to contact the RIS server.

   Troubleshooting: If the client does not get past the DHCP message, the client is not receiving a response from the remote installation server. Check the following possibilities:

   - Is the remote installation server available and has the (BINLSVC) RIS service started? RIS servers must be authorized in the Active Directory for their services to start. To ensure that the service has started, use the DHCP snap-in (click Start, point to Programs, point to Administrative Tools, and click DHCP).

- Are other remote boot-enabled clients receiving the Client Installation wizard? If so, this may indicate this client computer is not supported or is having remote boot ROM-related problems. Check the version of the PXE ROM on the client computer.
- Is a router between the client and the remote installation server not allowing the DHCP-based requests/responses through? When the RIS client and the RIS server are on separate subnets the router between the two systems must be configured to forward DHCP packets to the RIS server. This is because RIS clients discover a RIS server by using a DHCP broadcast message. Without DHCP forwarding set up on a router, the clients' DHCP broadcasts will never reach the RIS server. This DHCP forwarding process is sometimes referred to as DHCP Proxy or IP Helper Address in router configuration manuals.

  To verify DHCP set up, click Start, point to Programs, point to Administrative Tools, and click Event Viewer. Refer to your router instructions for setting up DHCP forwarding on your specific router.
- Are any error messages in the event log under the System or Application logs specific to RIS (BINLSVC), DNS, or the Active Directory?

3. The client changes to BINL or prompts the user to click the F12 key. This means that the client has contacted the RIS server and is waiting to TFTP the first image file-OSChooser. You might not see the BINL and TFTP message, because on some machines this sequence simply flashes by too quickly. (Note: Pressing F12 Key is automated, by swapping startrom.com and startrom.n12 files under <reminst_share>\OSChooser\i386 folder.

   Troubleshooting: If the client machine does not get a response from the Remote Installation Server, the client times out and displays an error that it did not receive a file from either DHCP, BINL, or TFTP. In this case, the RIS Server did not answer the client computer. Stop and restart the BINLSVC. From the Start menu, click Run, and type CMD. Enter these commands: Net Stop BINLSVC Net Start BINLSVC

   If the client machine does not receive an answer after attempting to stop and restart the service, check the Remote installation Server Object properties to ensure the correct setting has been set. Verify that RIS is set to "Respond to client computers requesting service", and "Do not respond to unknown client computers". Click Start, point to Programs, point to Administrative Tools, and click Event Viewer to check the Event log on the RIS server for any errors relating to DHCP, DNS, or RIS (BINLSVC).
4. At this point, the client should have downloaded and displayed the Client Installation wizard application with a Welcome screen greeting the user.

Problem: Is the Pre-Boot portion of the PXE-based Remote Boot ROM Secure?

Solution: No. The entire ROM sequence and OS installation/replication is not secure with regard to packet type encryption, client/server spoofing, or wire sniffer based mechanisms. As such, use caution when using the RIS service on your corporate network. Ensure that you only allow authorized RIS servers on your network and that the number of administrators allowed to install and or configure RIS servers is controlled.

Problem: While booting from the network, the target host displays the following error message:

```
No proxyDHCP offers were received.
```

Solution: The client machine/target host is not able to obtain an IP address from the DHCP server. For more details, see Step 2 above. See the following Microsoft knowledge base articles:

- Your RIS client may not be able to obtain an IP address during the PXE restart on Windows Server 2003 and Windows 2000
- Using Dynamic Host Configuration Protocol Options 60, 66, 67 to Direct PXE Clients to RIS Servers May Fail
- PXE Clients Do Not Receive IP Address From DHCP Server Across a Router

Problem: How do PXE Client, DHCP and RIS server interact?

Solution: See the following Microsoft knowledge base article: Description of PXE Interaction Among PXE Client, DHCP, and RIS Server .

Problem: Target host displays the following message while booting from the network:

```
ARP Timeout message
```

Solution: You see this error message when the client machine gets a valid IP address from the DHCP server, but invalid PXE Boot server IP address (RIS server's IP address in the provisioning subnet) from the BINL service on the RIS server. This is observed on some old machines like HP-Lpr when they are run as multi-homed RIS servers. However, this problem does not occur on newer hardware like Hp-Proliant DL 360 G3 series of server machines from the same vendor, even when they are configured as multi-homed RIS servers. To enable old machines like HP-Lpr's to work as RIS servers without displaying this error nessage, make sure the machines are not multi-homed. In other words, the system should have only one enabled interface which is in the provisioning subnet.

For more information, see the following Microsoft knowledge base article: A multi-homed RIS server may not answer all clients, and you may receive an error message on PXE clients that are running Windows Server 2003 or Windows 2000.

Problem: Text-mode installation does not boot.

Solution: Try the following solutions:

- Check the RIS image distribution setup.
- Check if there are any missing network adapter and mass storage drivers bundled with the image.
- Check RIS setup Information file for drivers path.
- Check CIW files (welcome.osc/login.osc/oschoice.osc/install.osc/warning.osc) under (<reminst share>/OSChooser/English) path for any invalid user information or missing AUTOENTER functionality.

- Change InitialBootTimeout in the ris properties file to a larger value. See Section RIS Drivers/Service Packs for more information on driver related issues. See Section GUID for more information on guid related issues.

Problem: GUI-mode installation goes into interactive mode.

Solution: Try the following solutions:

- Check RIS Setup Information file regarding data in the following sections: Unattended, UserData, GuiRunOnce, and GuiUnattended.
- Make sure you provide the default Administartor's password and Product Key in the sif files. See Section RIS CIW/SIF Files for more information on CIW and ris SIF files.

Problem: How to change default timeout values for text-mode installation and GUI-mode installation for each client?

Solution: Before starting the provisioning activity, make sure you change the default timeout values for the following properties in the ris.properties file (usually located under <n1osp folder>/etc/) on your N1 OS provisioning server.

```
ris.InitialBootTimeout
ris.OsInstallTimeout
```

Issues Related to Remote Information Services (RIS)

Problem: How to Enable Debug Mode for Remote Install Servers?

Solution: Follow the instructions as described in the Microsoft knowledge base article 236033.

Problem: How do you automate the CIW screens for RIS services?

Solution: See the following Microsoft knowledge base articles:

- Description of Client Installation Wizard Screens for Remote Installation Services - 268325
- HOW TO: Automate CIW Screens using AUTOENTER - 824184

Problem: Where can I find more information on Setup Information Answer files (.sif files)?

Solution: See the `deploy.cab` file on Windows 2000/2003 Server Resource Kit CD for more details.

Problem: How do you change the Administrator Password during RIS Installation?

Solution: See the following Microsoft knowledge base article: How to Set the Administrator Password During RIS Installation - 257948.

Problem: How do you add drivers to a RIS Image?

Solution: See the following Microsoft knowledge base articles:

HOW TO: Add Third-Party OEM Network Adapters to RIS Installations - 246184

HOW TO: Add OEM Plug and Play Drivers to Windows Installations - 254078

Problem: How do you slipstream a service pack into a RIS image?

Solution: See the following articles on the Microsoft web site:

- How to Obtain the Latest Windows 2000 Service Pack
- Slipstream Switch for Windows 2000 SP1 Update.exe Does Not Work with RIS Server Images
- Information on how to create a RIS image with the service pack bundle is available in the Service Pack 3 Installation and Deployment Guide. Similar guides can be found for other service packs as well on the service packs web site.
- Windows 2000 Service Pack 3 Express/Network Installation. Similar download pages can be found for other service packs as well.

Note – Choose "Network installation" for downloading a service pack to create a RIS image.

Problem: You see the following error message during text-mode installation:

```
Illegal or Missing File Types Specified in Section SCSI.Name
```

Solution: See the Microsoft knowledge base article 275334.

Problem: You see an error message during text-mode installation when you try to install a RIS image. The error message includes:

```
Setup Cannot Continue
```

Solution: See the Microsoft knowledge base article 830751.

Problem: You see the following error message during text-mode installation:

```
INF File Tmp\<GUID_number.sif> Is Corrupt or Missing
```

Solution: See the Microsoft knowledge base article 224830.

Problem: You see the following error message during text-mode installation:

```
The Operating System Image You Selected Does Not Contain the Necessary Drivers
```

Solution: See the Microsoft knowledge base article 247983.

Problem: You see the following error message during text-mode installation:

```
The Operating System Image You Selected Does Not Contain the Necessary
Drivers for Your Network Adapter. Try Selecting a Different Operating System
Image. If the Problem Persists, Contact Your System Administrator.
```

Solution: See the Microsoft knowledge base article 315074.

Problem: The remote install client hangs at the end of text-mode set up.

Solution: See the Microsoft knowledge base article 226941.

Problem: The RIS set up stops responding on the “Setup is Starting Windows” screen.

Solution: See the Microsoft knowledge base article 320865.

Issues Related to GUID

Problem: Where do I look on the client computer to find the GUID/UUID for pre-staging clients in the Active Directory for use with RIS?

Solution: The GUID/UUID for client computers that are PC98 or Net PC compliant can be found (in most cases) in the system BIOS. OEMs are encouraged to ship a floppy disk containing a comma-separated file or spreadsheet that contains a mapping of serial number to GUID/UUID. This allows you to script pre-staging client computers within the Active Directory. OEMs are also encouraged to post the GUID/UUID on the outside of the computer case for easy identification and pre-staging of computer accounts. If the GUID is not found in the above-mentioned locations, you can sniff the network traffic of the client to locate the DHCP Discover packet. Within the DCHP Discover packet, you can find the 128-bit 32 byte GUID/UUID.

Problem: Two client machines have the same GUID value.

Solution: RIS fails in this case, because RIS identifies each target host as a computer object in its active directory with an unique GUID value. If multiple objects have the same GUID, the RIS client machine throws an error during its setup phase. You see the following message:

```
BINLSVC found Duplicate GUID accounts on the RIS Server.
Please contact your system Administrator.
```

To overcome this issue, delete any old computer accounts with the same GUID in the RIS server’s Active Directory before proceeding further.

CHAPTER 3

OS Provisioning Deployment Environment

This chapter provides guidelines for setting up an environment that supports OS provisioning.

Prerequisites

Provisioning an operating system using the OS provisioning plug-in requires that you have a basic understanding of system administration and networking. In addition, to provision the operating system requires that basic IP connectivity exists between the machines.

Basic OS Provisioning Environment

The basic OS provisioning environment has the following requirements:

- N1 Grid SPS Master Server — A system upon which the N1 Grid Service Provisioning System software is installed and that is configured to run as a Master Server.
- OS provisioning server — A Solaris 9 9/04 (or later) system upon which the OS provisioning server is installed through the plug-in. The OS provisioning server runs DHCP services to respond to target host requests.
- Solaris boot and install server — To provision Solaris systems, you need a Solaris boot and install server on which Solaris 9 9/04 or later is installed.
- Linux boot and install server — To provision Linux systems, you need a Linux boot and install server on which Linux is installed.

- Windows boot and install server — To provision Windows systems, you need a Windows boot and install server on which Windows 2003 is installed.
- The OS provisioning server, boot and install servers, and provisioning targets must be accessible in a network through a native LAN, a VLAN or through routers. Refer to the appropriate networking and operating system documentation for more information about configuring your network.
- DHCP packets from the provisioning targets must be able to reach the OS provisioning server.

Note – The N1 Grid SPS Master Server, OS provisioning server, and Solaris boot and install server can be one physical system. However, running all three servers on one system increases the load on the server and increases the network traffic that the server has to handle. Keeping them separate enables you to scale better in the future.

Target Hosts

You need to set up provisionable target systems for OS provisioning. The OS provisioning server needs to know information about these targets, such as MAC address, GUID, remote management connections, and access information. For information about defining targets, see Chapter 8.

Network

The OS provisioning plug-in is designed to work with a wide range of network configurations and topologies. As such, the plug-in does not dictate any network topology nor does it manipulate network elements like switches or routers for its needs. However, the plug-in relies on the existence of some network communication:

- Layer 2 connectivity between the provisioning interface of the OS provisioning server and the provisioning interface of the target host
- IP connectivity between the provisioning interface of the boot and install server and the provisioning interface of the target host
- IP connectivity between the OS provisioning server, the network management port of the target host, and the control network interface of the boot and install server

These requirements on the networking infrastructure are imposed by the needs of the two network types central to the function of the OS provisioning server. Those network types are the *control network* and the *provisioning network*.

Note – An *access network* is the network used to access the OS provisioning and boot and install servers. An example of an access network is the corporate intranet. This network is not needed for OS provisioning functionality. From a security standpoint, you should keep the access network separate from the control and provisioning networks.

The following diagram illustrates the network environment.

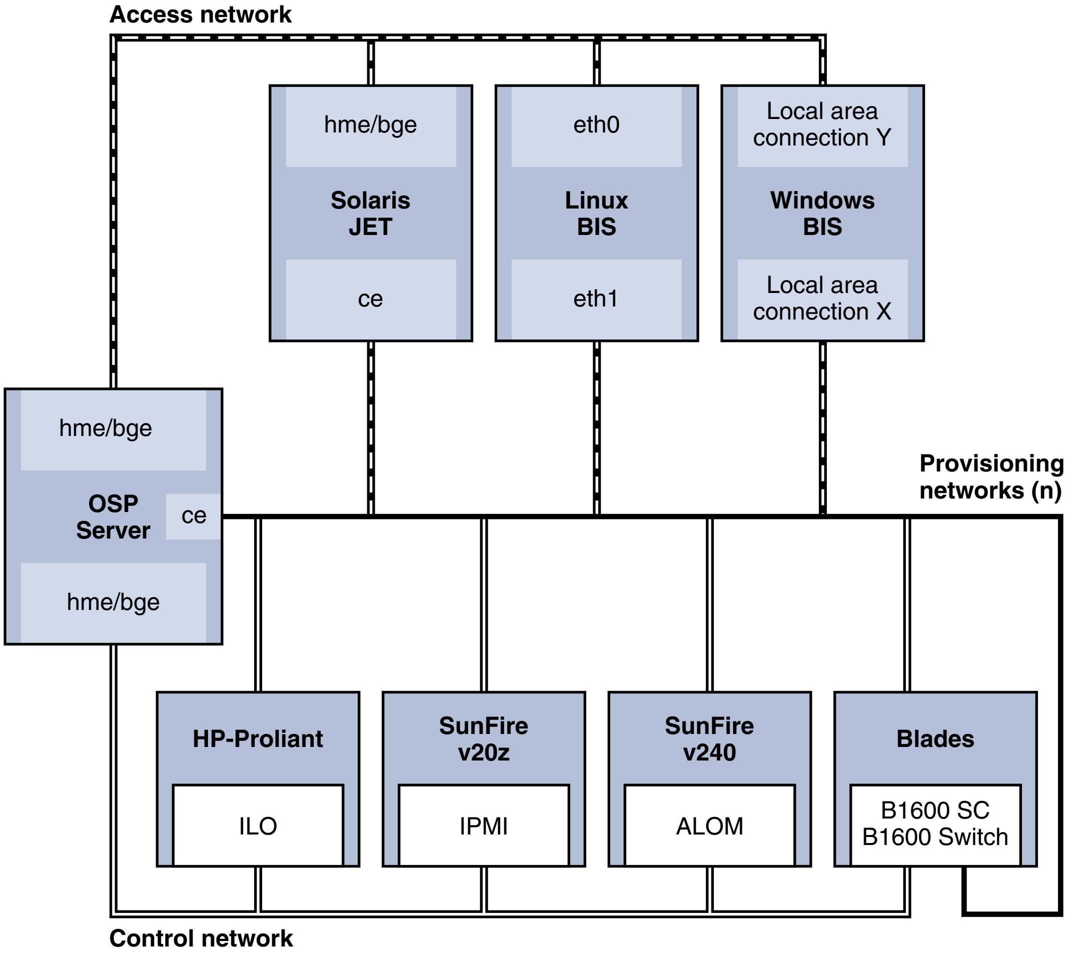

FIGURE 3–1 Network Environment Diagram for OS Provisioning

Provisioning Network

A provisioning network is comprised of the provisioning interface of the OS provisioning server, the provisioning interfaces of the target platforms, and the provisioning interfaces of one or more boot and install servers. The provisioning network can be comprised of one or more subnets. An OS provisioning plug-in installation supports the use of multiple provisioning networks for OS provisioning. The protocols and technologies that are required for network-based provisioning dictate the requirements of these provisioning networks. These requirements are:

- The provisioning interface of the OS provisioning server and the provisioning interfaces of the target platform should be in a common Layer 2 broadcast domain,
- The boot and install servers provisioning interface serving the host should have IP connectivity to the subnet of the target platforms' provisioning interface. This can be achieved by one of two ways:
  - The boot and install server has a network interface in the provisioning subnet,
  - The boot and install server has a provisioning interface that is reachable from the provisioning subnet by means of routing.

Control Network

The control network is the network used by the OS provisioning server for two primary functions:

- Managing the network management port of the target host to control power state, boot order, and console of the target platform
- Configuring the boot and install servers for provisioning activity

The control network can be a pure IP network or may have serial/terminal server elements. The OS provisioning server communicates with the boot and install servers over an IP network. At the same time, communication with the network management port of the target host may occur over an IP network or a serial network. The control network can span many subnets. The only requirement on the control network is that all boot and install servers and target network management ports can be routed from the OS provisioning server.

Switched Networks

The above requirements take on special meaning in a switched environment. In a switched network, the switched connections can be in either trunk or access (non-trunk) modes. For the control network, switched connections can be in access mode because IP routing from the OS provisioning server is all that is required. The provisioning network can have switched ports in either trunk or access modes depending on the provisioning network design.

Security

The OS provisioning plug-in software leverages the N1 Grid SPS security model. Most communication between the different servers occurs through the N1 Grid SPS Remote Agents (RAs). Configure the RAs for secure communication. See documents for more information on how to enable secure communication between the Master Server and the RAs.

For remote management of the targets, the encrypted passwords are stored on the OS provisioning server. For information about encrypting the passwords, see "Password Encryption" on page 113.

For communication with the Windows boot and install server, you need to activate either RSH or SSH services. Use SSH services to secure communications between the OS provisioning server and Windows boot and install server. For information, see "How to Install Windows SSH Server on the Windows RIS Server" on page 82.

Configuring New Environments

The Sun Data Center Reference Architecture captures and applies best practices to define a generic data center configuration. This architecture can then be reliably and quickly assembled, tested, and deployed with lower risk and lower total cost of ownership (TCO). Data Center Reference Architecture Implementations are instantiations of the Sun Data Center Reference Architecture that provide complete details with actual hardware and software products and technologies, along with services, to meet customer requirements. Data Center Reference Architecture Implementations are pre-designed, pretested groups of components for small, medium, and large data centers, and provide a production-ready target environment for enterprise consolidation and migration projects.

The Sun Data Center Reference Architecture Implementation framework is a flexible combination of SunFire Servers, Sun StorEdge™ storage arrays, Sun Java™ Enterprise System and Solaris software, as well as LAN and SAN infrastructure. For more information, see the Sun Data Center Reference Architecture web site.

Process Overview

1. Prepare the hardware for the N1 Grid SPS Master Server, OS provisioning server, and boot and install servers.
2. Obtain the N1 Grid SPS software.
3. Install the N1 Grid SPS Master Server, as explained in "Installing the N1 Grid Service Provisioning System 5.0" in *N1 Grid Service Provisioning System 5.0 Installation Guide*.

4. Install the N1 Grid SPS RA and N1 Grid SPS command-line interface (CLI) on the OS provisioning server
5. Install the N1 Grid SPS RA and N1 Grid SPS CLI on the Solaris boot and install server
6. Install the N1 Grid SPS RA on the Linux boot and install server
7. Prepare the RAs on the OS provisioning server, Solaris boot and install server, and Linux boot and install server. For information, see "How to Prepare a Physical Host" in *N1 Grid Service Provisioning System 5.0 System Administration Guide*.

Note – For safety, back up the N1 Grid SPS database. For information, see Chapter 9, "Backing Up and Restoring," in *N1 Grid Service Provisioning System 5.0 System Administration Guide*.

▼ How to Enable the Master Server to Use Session IDs

Steps

1. **Edit the Master Server configuration file.**

   By default, this file is located at the following location:

```
/opt/SUNWn1sps/N1_Grid_Service_Provisioning_System_5.0/server/config/config.properties
```

2. **If this is an existing N1 Grid SPS installation, follow these steps:**

   a. **Find the session ID entry that looks similar to the following:** **`config.allowSessionIDOnHosts=masterserver,biss1`**

   b. **Change the value after the equals sign to the names of the OS provisioning server and the Solaris boot and install server.**

      For example: `config.allowSessionIDOnHosts=myspsserver,sol10bis`

3. **If this is a new N1 Grid SPS installation, add a line similar to the following:** **`config.allowSessionIDOnHosts=masterserver,biss1`**

   The value after the equals sign must include the names of the OS provisioning server and the Solaris boot and install server.

4. **Adjust global plan execution timeouts for your environment.**

   Change the following entries in the `config.properties` file:

```
pe.defaultPlanTimeout=12000
pe.nonPlanExecNativeTimeout=12000
```

   Where the timeouts are in seconds. Timeout should be greater than the longest plan run operation that you expect for your site. The default plan timeout is 30 minutes (1800 seconds). The default native timeout is 10 minutes (600 seconds).

The above example shows an arbitrary higher timeout value of 200 minutes (12000 seconds).

5. **To enable these changes, stop and restart the Master Server.**

   Login to the Master Server as `n1sps` and type the following commands:

   ```
   # cr_server stop
   # cr_server start
   ```

   By default, these commands are in the following file:

   ```
   /opt/SUNWn1sps/N1_Grid_Service_Provisioning_System_5.0/server/bin
   ```

Configuring Existing Environments

You can use the OS provisioning plug-in to provision the OS in an existing server and network environment. The following paragraphs describe in detail how you could use the plug-in in an existing environment

Hardware and Software Configuration

Ensure that you have hardware to support the N1 Grid SPS Master Server, OS provisioning server, Solaris boot and install server, Linux boot and install server, and Windows boot and install server. See "Supported Systems" on page 20 for information about appropriate systems.

Network Environment

Verify that the Master Server, OS provisioning server, and boot and install servers are able to connect with each other through an IP network.

Ensure that you have enough bandwidth to provision the servers simultaneously. The bandwidth requirements vary depending on how many simultaneous provisioning operations you intend to perform.

Note – Simultaneous OS installations require lot of bandwidth and might experience failures or timeouts if the bandwidth is not available. To avoid problems, either physically separate the traffic or deploy more boot and install servers.

DHCP Services

The OS provisioning server uses its own DHCP service. The DHCP service is used during the provisioning operation to provide install time parameters and install time IP addresses to targets. The DHCP service does not respond to clients that are not being provisioned. Therefore, if you have other DHCP services serving in this subnet, ensure that these services are not responding to the targets during the provisioning operation. Once the OS has been provisioned, you can reactivate DHCP to respond to the targets. Ensure that the target DHCP packets can reach the OS provisioning server by either locating the OS provisioning server in the same subnet or through routing.

Target Hosts

The OS provisioning plug-in can automate the power off and power on cycles during provisioning. Enable the remote management interfaces (if any) of the targets. If the target does not support remote management, use the generic target. For more information about target hosts, see Chapter 8

N1 Grid Service Provisioning System Software

Ensure that the N1 Grid SPS software is version 5.0 or later.

Existing Solaris JET Environments

If you are running the JumpStart Enterprise Toolkit (JET) technology, that product must be uninstalled before you can use the OS provisioning plug-in. For more information , see "Setting up the Solaris JET Server" on page 55

The default base directory of the SUNWjet package that ships with the OS provisioning plug-in is `/opt/SUNWjet`. Earlier versions of SUNWjet used the `/opt/jet` default base directory.

If you are using an existing JET package, uninstall the existing package, then create the JET server, as explained in "Setting up the Solaris JET Server" on page 55. This process performs the following tasks:

1. Installs the version of SUNWjet included with the OS provisioning plug-in at `/opt/SUNWjet`.
2. Creates symbolic links between any pre-existing JET product modules in `/opt/jet/Products` over to the `/opt/SUNWjet/Products` location.

Once the process completes, when you use the OS provisioning plug-in to create new Solaris profiles, you can include by name any JET product modules that were installed previously on the server.

The previous /opt/jet/Templates and /opt/jet/Clients areas are left untouched. You can then refer to those areas as needed, in case some of their values are helpful for creating new Solaris profiles with the OS provisioning plug-in.

CHAPTER 4

Installing and Configuring the OS Provisioning Plug-In

From an operating system (OS) provisioning perspective, installation consists of three phases:

- Installing the N1 Grid SPS software, , as explained in *N1 Grid Service Provisioning System 5.0 Installation Guide*
- Installing the OS provisioning plug-in
- Installing the operating system environment that you want to provision

All three installation phases likely also have configuration steps.

This chapter includes the following information:

Installing the OS Provisioning Plug-In

Acquiring the OS Provisioning Plug-In

The OS provisioning plug-in is packaged as a *plug-in* to the N1 Grid SPS software. Plug-ins are packaged in Java™ Archive (JAR) files. The plug-in files for the OS provisioning plug-in are available from the N1 Grid SPS Supplement CD or from the Sun Download Center.

Adding the OS Provisioning Plug-In to the N1 Grid SPS

To make a given plug-in known to the N1 Grid SPS product, you need to import the plug-in. To import a plug-in, follow these steps as explained in detail in Chapter 5, "Plug-In Administration," in *N1 Grid Service Provisioning System 5.0 System Administration Guide*.

1. In the Administrative section of the browser interface main window, click Plug-ins.
2. In the Action column of the Plug-ins page, click Import.
3. Browse to the location where you downloaded the `com.sun.isp_1.0.jar` file.
4. Click the Continue to Import button.

When the import completes successfully, a plug-in details page appears that shows you the objects that the plug-in provides.

You can also import a plug-in archive file from the command line. Use the following command:

```
% cr-cli -cmd plg.p.add -path com.sun.isp_1.0.jar -u username -p password
```

Creating the OS Provisioning Server

Although the OS provisioning plug-in has been imported, you cannot provision an operating system until you create and set up an OS provisioning server.

▼ How to Create the OS Provisioning Server (Browser Interface)

At the end of this task, the OS provisioning software is installed on the OS provisioning server. In addition, the DHCP service is installed on the OS provisioning server.

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the OSP Control Server section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

- **To use an existing variables set, select a name from the drop-down menu in the Service component row of the Plan Parameters table.**
- **To create a new variables set, click Select from List in the Service component row of the Plan Parameters table.**

  a. **Click Create Set.**

  b. **Type a name for the variables set.**

  c. **Change variables as needed.**

Variable Name	Description
`installPath`	Base directory into which to install the OS provisioning software.
`vhost_name`	Name of the virtual host that represents the OS provisioning server.
`host_dir`	Path to directory in which target host files reside.
`profile_dir`	Path to directory in which OS profile resides.
`subnet_dir`	Path to directory in which subnet files reside.
`logs_console_dir`	Path to directory in which logs and console files reside.

  d. **Save the variables set.**

  e. **Select the variables set that you just saved from the drop-down menu in the Service component row of the Plan Parameters table.**

5. **Select the host on which you want to create the OS provisioning server.**

6. **Click Run Plan (includes preflight).**

   This plan takes some time to run, usually approximately 15 minutes. While the installation is proceeding, click on the Details links in the Plan Run window to see progress.

7. **To verify that the OS provisioning server is created successfully, click the Hosts link in the left side of the provisioning server window.**

   You should see the name of the virtual host appended with `-osp`. For example, if you installed on host `masterserver`, the virtual host is `masterserver-osp`.

▼ How to Create the OS Provisioning Server (Command-Line Interface)

Step ● **To create the server from the command line, type a command similar to the following example:**

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID "NM:/com/sun/n1osp/untyped/Service-create" \
-tar H:NM:masterserver  -comp + -vs + -pto 30 -nto 10
```

Note – The target should be the OS provisioning server. In the example, the N1 Grid SPS Master Server is used as the OS provisioning server.

Creating and Registering the Subnet

The OS provisioning server provides DHCP service. The DHCP service has to listen on all the subnets that will be used to provision the targets. In the N1 Grid SPS interface, you need to identify the subnets to be used for provisioning targets so that the OS provisioning server can respond to DHCP requests. The OS provisioning service will create the interfaces with the required addresses on the provisioning server (if not already created) during the provisioning operation.

▼ How to Identify the Subnet for the OS Provisioning Server (Browser Interface)

To manage the subnet from the browser interface, follow these steps.

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**
2. **On the OS Provisioning Common Tasks page, click Manage in the OSP Subnets section.**
3. **On the Component Details page, click the Run action in the Create row.**
4. **Choose variables to use for this plan.**
   - **To use an existing variables set, select a name from the drop-down menu in the `Service` component row of the Plan Parameters table.**
   - **To create a new variables set, click Select from List in the `Service` component row of the Plan Parameters table.**

a. **Click Create Set.**

b. **Type a name for the variables set.**

c. **Change variables as needed.**

Variable	Description	Example
`installPath`	Subnet address	`10.42.42.0`
`mask`	Subnet mask	`255.255.255.0`
`gateway`	Gateway for the subnet	`10.42.42.1`
`host_interface`	Interfaces on the OS provisioning server to be used for this subnet	`hme0`
`host_address`	Host address that is assigned to the `host_interface`	`10.42.42.1`

Note – The OS provisioning software creates the interfaces and assigns the addresses during provisioning operation, if those values are not set already.

d. **Save the variables set.**

e. **Select the variables set that you just saved from the drop-down menu in the `Service` component row of the Plan Parameters table.**

5. **Select the virtual host to which this subnet applies.**

Tip – The virtual host for the OS provisioning server ends in `-osp`.

6. **Click Run Plan (includes preflight).**

▼ How to Identify the Subnet for the OS Provisioning Server (Command-Line Interface)

To manage the subnet from the command line, follow these steps:

Steps 1. **Create the variables set.**

Type a command similar to the following example:

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/Subnet -name "subnet1" \
-u admin -p admin -vars "installPath=10.42.42.0;mask=255.255.255.0;gateway=10.42.42.1; \
host_interface=ce8000;host_address=10.42.42.1"
```

For more information about the variables, see Step 4 in "How to Identify the Subnet for the OS Provisioning Server (Browser Interface)" on page 50.

2. **Run the plan to create the subnet using the variables set created in the previous step.**

   Type a command similar to the following example:

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/n1osp/untyped/Subnet-create  \
-tar H:NM:masterserver-osp -comp + -vs subnet1 -pto 30 -nto 10
```

▼ How to Add a New Provisioning Subnet

You can use several subnets to provision operating systems. The following steps explain how to add a new subnet to the OS provisioning server.

Before You Begin Verify that the OS provisioning server, the boot and install server for the applicable operating system, and the target host are in the same IP network.

Steps

1. **Create a new subnet component as described in "How to Identify the Subnet for the OS Provisioning Server (Browser Interface)" on page 50.**

   For IP connectivity, you have two options:

   - Create new IP addresses for the OS provisioning server .
   - Use existing IP addresses, but create a route for DHCP packets to reach from the target to the OS provisioning server.

2. **Create new interfaces on the boot and install servers for the new subnet or define routes so that the boot and install server can be reached by the target.**

3. **Verify that the file system shares are updated to deliver the OS media in this new subnet.**

4. **Verify that the targets are in this subnet.**

5. **Edit the profile you want to provision for IP addresses to be in the new subnet.**

CHAPTER 5

Provisioning the Solaris Operating Environment

This chapter explains how to use the OS provisioning plug-in to install the Solaris Operating System onto target hosts.

The chapter contains the following information:

JumpStart Enterprise Toolkit (JET) Technology

The OS provisioning plug-in uses the features provided through the JumpStart™ Enterprise Toolkit (JET) technology to provision the Solaris Operating System (OS). JET is an enhancement to the Sun-developed JumpStart technology that automates the installation of the Solaris OS over a network. The OS provisioning plug-in through JET enhances this capability further, while hiding some of the complexity.

JET Modules

JET technology provides the JumpStart server with product-specific modules that install the Solaris OS and other products in a structured way. This structure enhances the features that you can implement through "ad-hoc" scripting of the JumpStart finish script.

The OS provisioning plug-in supplies three JET modules:

- `base_config` – Installs and configures the Solaris OS. For information about `base_config` variables, see "Basic OS Configuration Variables" on page 129.
- `spsra` – Installs and configures the N1 Grid SPS Remote Agent (RA) on a Solaris system. For information about `spsra` variables, see "Variables for N1 Grid SPS Remote Agents" on page 138.
- `custom` – Installs arbitrary lists of Solaris packages, patches, and files, and can run arbitrary collections of scripts. For information about custom modules, see "Using the `custom` Module" on page 65.

Solaris Installation Process through JET

The build sequence of the JumpStart Enterprise Toolkit is as follows:

1. Standard Solaris installation phase
2. Call standard JumpStart finish script
3. Call individual module "install" scripts
4. Reboot target server
5. (Optional) Platform related installation tasks; reboot after each level
6. (Optional) Application related installation tasks; reboot after each level
7. (Optional) Final installation tasks (no reboot)
8. login prompt appears on the console

The optional steps after the initial reboot depend on the individual modules configured within the target server template. Modules can be written in such a way that they request that the toolkit perform additional work after the first reboot. In this request, the modules can identify whether the work should be in the platform-related area, the application-related area, or whether the work needs to be done at the end, when no more reboots are planned.

Solaris Provisioning Process Overview

1. Set up the JET server.
2. Create the Solaris image or attach an existing image to the JET server.
3. Create a target host.
4. Install the image onto the target host.

Setting up the Solaris JET Server

The Solaris boot and install server is a JET server.

▼ How to Create the JET Solaris Image Server

You can create a JET image server from the N1 Grid SPS browser interface or command-line interface. To create the server from the command line, type the following command:

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/n1osp/untyped/Jet-create \
-tar H:NM:biss1 -comp + -vs + -pto 30 -nto 10
```

To create the server from the browser interface, follow these steps.

Steps

1. **Log into the system that is to be the Solaris boot and install server.**
2. **If you currently use the JumpStart enterprise toolkit, remove those files.**

   Use the following command: # **`pkgrm SUNWjet`**
3. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**
4. **On the OS Provisioning Common Tasks page, click Create in the JET Solaris Image Servers section.**
5. **On the Plans Details page, click Run.**
6. **Choose variables to use for this plan.**
   - **To use an existing variables set, select a name from the drop-down menu in the JET component row of the Plan Parameters table.**
   - **To create a new variables set, click Select from List in the JET component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Verify variables and change them, if needed.**

     The image server component has only a few variables, most of which you are unlikely to change. See "Solaris Image Server Component Variables" on page 56 for a list of those variables and their default values.

     d. **Save the variables set.**

e. **Select the variables set that you just saved from the drop-down menu in the JET component row of the Plan Parameters table.**

7. **Select the host on which to create the JET image server.**

8. **Click Run Plan (includes preflight).**

9. **To verify that the JET image server is created successfully, click the Hosts link in the left side of the provisioning server window.**

   You should see the name of the virtual host appended with `-jet`. For example, if you installed on host `masterserver`, the virtual host is `masterserver-jet`.

Solaris Image Server Component Variables

The following variables apply to the image server component.

TABLE 5–1 Solaris Image Server Component Variables

Variable Name	Description	Default Value
`installPath`	Location to install JET package.	`/opt/SUNWjet`
`sps_cli`	Location of N1 Grid SPS command-line interface.	`/opt/SUNWn1sps/N1_Grid_Service_Provisioning_Syst` `cli/bin/cr_cli`
`jetHost`	Name to use for the virtual host name. By default, the virtual JET host name is set to the host name of the target host with `-jet` appended.	`:[target]-jet`
`jetFolder`	Folder where you want to store JET-specific information. By default, the JET folder is located in `/com/sun/n1osp/autogen-`*:[jetHost]*, where *:[jetHost]* is the virtual JET host name defined in the previous step.	`/com/sun/n1osp/autogen-:[jetHost]`
`templatePrefix`	Value to use to identify the OS provisioning JET templates.	`OSP_TEMPLATE_`

Creating Solaris Images and Profiles

To provide basic JumpStart functionality, an image of the appropriate Solaris OS media must be installed on the JET server. That image must then be attached to a profile that explains how the image is to be installed.

▼ How to Import a Solaris Image

To provide basic JumpStart functionality, an image of the appropriate Solaris OS media must be installed on the JET server. The version or versions of Solaris to be imported onto the JET server depend on those required by the target servers. You can install multiple versions of the Solaris media on the same JET server at the same time.

Note – While it is recommended that you copy the media to disk on the JET server, it is not strictly required. You can share the physical media appropriately to allow the target servers to boot from it. However, using the physical media severely restricts the performance of the target server build and minimizes your ability to support multiple versions of Solaris.

Steps

1. **Mount the Solaris DVD on the JET server, either through a local drive, or by using a shared drive on the network.**

2. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

3. **On the OS Provisioning Common Tasks page, click Import in the Solaris Images section.**

4. **On the Plans Details page, click Run.**

5. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the SolarisImage component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the SolarisImage component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Type the Solaris main version number into the `version` variable.**

     d. **Type the Solaris minor version number into the `release` variable.**

     e. **Verify the remaining variables and change them, if needed.**

     The following table describes the additional variables and their default values.

Variable Name	Description	Default Value
`architecture`	Architecture of the physical system, for example, SPARC or x86.	`sparc`
`installPath`	Unique name for the imported Solaris image.	`Solaris:[version]_:[release]_:[architecture]` For example: `Solaris9_u7_sparc`
`image_dir`	Path to where imported image will be stored.	`/export/osp_image`
`image_subnet_addr`	Subnet address of the image server.	`:[target(/):sys.ipAddress]`
`image_subnet_mask`	Subnet mask of the image server	`255.255.255.0`
`media_src`	Path to the image files. By default, the software assumes the files are on a DVD disc.	`/cdrom/cdrom0/s0`

f. **Save the variables set.**

g. **Select the variables set that you just saved from the drop-down menu in the SolarisImage component row of the Plan Parameters table.**

6. **On the Plan Details Run page, select the JET image server on which to import the image.**

Tip – The host name of the JET image server ends in `-jet`.

---

7. **Click Run Plan (includes preflight).**

▼ How to Create a Solaris Provisioning Profile (Browser Interface)

Before You Begin Make sure the JET server exists and that the Solaris OS software is available to that server. For more information about creating the JET server, see “Setting up the Solaris JET Server” on page 55. For more information about creating the Solaris image, see “How to Import a Solaris Image” on page 57.

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**
2. **On the OS Provisioning Common Tasks page, click Create Profile in the Solaris Images section.**
3. **On the Plans Details page, click Run.**
4. **Choose variables to use for this plan.**
   - **To use an existing variables set, select a name from the drop-down menu in the SolarisImage component row of the Plan Parameters table.**
   - **To create a new variables set, click Select from List in the SolarisImage component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Type the Solaris main version number into the `version` variable.**

     d. **Type the Solaris minor version number into the `release` variable.**

     e. **Verify the remaining variables and change them, if needed.**

     The following table describes the additional variables and their default values.

Variable Name	Description	Default Value
`architecture`	Architecture of the physical system, for example, SPARC or x86	`sparc`
`installPath`	Solaris image name to attach	`Solaris:[version]_:[release]_ :[architecture]` For example: `Solaris9_u7_sparc`
`image_dir`	Path to location where image file resides	`/export/osp_image`
`image_subnet_addr`	Subnet address of the image server	`:[target(/) :sys.ipAddress]`
`image_subnet_mask`	Subnet mask of the image server	`255.255.255.0`

     f. **Save the variables set.**

   g. **Select the variables set that you just saved from the drop-down menu in the SolarisImage component row of the Plan Parameters table.**

5. **On the Plan Details Run page, select the JET image server on which to create the profile.**

Tip – The host name of the JET image server ends in `-jet`.

---

6. **If needed, change the Profile Name.**

7. **If needed, change the Profile Description.**

8. **Update the list of the JET modules to use.**

   The JET product modules are located on the JET server in the directory `/opt/SUNWjet/Products`. For more information about creating JET modules, see Appendix C.

9. **Click Run Plan (includes preflight).**

10. **(Optional) Change default profile values.**

    a. **Navigate to the Solaris folder that contains the new Provision component.**

    For example, `/com/sun/n1osp/autogen-biss1-jet/provision/`.

    b. **Click the Solaris provisioning component name.**

    For example, `Solaris10_0205_sparc.standard`.

    c. **Edit default values.**

    i. **Click the Edit button at bottom of Details page.**

    ii. **Change values.**

    iii. **Click the Check-in button at the bottom of the Details page.**

▼ How to Create a Solaris Provisioning Profile (Command-Line Interface)

Steps 1. **To create a variable set for the SolarisImage component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/SolarisImage \
-name "solaris9" -u admin -p admin -vars "version=9;release=u7;architecture=sparc; \
installPath=Solaris_9;image_dir=/export/install;image_subnet_addr=10.42.42.2; \
image_subnet_mask=255.255.255.0;media_src="
```

2. **To provide JET module information, create a file `/tmp/solaris-image` that contains the following entries:**

   ```
   standard
   Standard Solaris
   base_config spsra
   ```

   Where:

   - The first line in the file corresponds to the Profile Name field in the plan variables section of the browser interface.
   - The second line in the file corresponds to the Profile Description field in the plan variables section of the browser interface.
   - The third line in the file corresponds to the JET Module Name field in the plan variables section of the browser interface.

3. **To run the plan, type a command similar to the following example:**

```
# cat /tmp/solaris-image|cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/untyped/SolarisImage-attach \
-tar H:NM:biss1-jet -comp + -vs solaris9 -pto 30 -nto 10
```

Installing the OS on the Target Host

The actual provisioning component contains a long list of variables that you might modify. A complete list of all available variables is in Appendix B.

▼ How to Install the OS on the Target Host (Browser Interface)

The following task includes a likely subset of variables for you to change when you install the OS onto a target host.

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Provision OS in the Solaris Images section.**

3. **Navigate to the Solaris folder that contains the `provision` component.**

   For example, `/com/sun/n1osp/autogen-biss1-jet/provision/`

4. **Select the Solaris provisioning component name.**

For example, `Solaris_9.standard`

5. **On the Component Details page, click the Run action next to the Provision_start:Install procedure.**

6. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the component name row of the Plan Parameters table.**

   - **To change values in an existing variables set, click Select from List in the component name row of the Plan Parameters table.**

     a. **Click the Edit button in the column for the variables set to change.**

     b. **Verify and change variables as needed.**

        A list of variables, their descriptions, and default values is in Appendix B.

     c. **When all values are correct, click the Save button.**

     d. **Select the variables set that you just saved from the drop-down menu in the component name row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the component name row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Verify or change variable values.**

        Although there are many variables that you might modify, you are likely to change only a few. The variable names and prompts in the Component Variables list should enable you to provide appropriate input for the variables. A list of variables, their descriptions, and default values is in Appendix B.

     d. **When all values are correct, click the Save button.**

     e. **Select the variables set that you just saved from the drop-down menu in the component name row of the Plan Parameters table.**

7. **Select the target host on which to provision the OS.**

   Tip – The target host should have a host name that ends in `-target`.

8. **Click Run Plan (includes preflight).**

   When the plan completes and the Solaris image is installed on the target host, you need to login to the target host. The password for the root user is `newroot`.

▼ How to Install the OS on the Target Host (Command-Line Interface)

Step ● **Type a command similar to the following example and replace the name of the component and target with your component name and target:**

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID \
NM:/com/sun/n1osp/autogen-biss1-jet/provision/SolarisProfile-provision-start-Solaris_9.standard \

-tar H:NM:testv240-target -comp + -vs + -pto 60 -nto 60
```

▼ How to Check OS Provisioning Status

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Status in the OS Provisioning Administration Tasks section.**

3. **On the Plans Details page, click Run.**

4. **On the Plan Details Run page, select the target host on which you provisioned the OS.**

5. **Select the target host on which to check the provisioning status.**

6. **Follow the Details links to view the status.**

▼ How to Stop an OS Provisioning Operation

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Abort in the OS Provisioning Administration Tasks section.**

3. **On the Plans Details page, click Run.**

4. **On the Plan Details Run page, select the target host on which you provisioned the OS.**

5. **Select the target host on which to stop the provisioning operation.**

JET Solaris Server Administration Tasks

▼ How to Import N1 Grid SPS RA Installers

To enable a cross-platform installation (for example, to install x86 platforms from a SPARC system), you need to import the architecture-specific installer to the JET Solaris Image server.

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**
2. **On the OS Provisioning Common Tasks page, click Manage in the JET Solaris Image Servers section.**
3. **On the Component Details page, click the Run action next to the import_product_media procedure.**
4. **On the Plans Run page, set the plan variables as follows:**

Variable	Description	Sample Value
JET Module Name	Name of JET module	`spsra`
Product Version	Version of N1 Grid SPS installer	`5.0`
Architecture	Architecture that applies to RA installer	`sparc`
Media Path	Path to directory that contains the RA installer	`/net/myothersys/export/ra`

5. **Click Run Plan (includes preflight).**

Note – To install from imported RA installers, the Solaris Profile used when installing the RA must have the value for “Install RA from snapshot (y,n)” set to `n`. See “Variables for N1 Grid SPS Remote Agents” on page 138.

▼ How to Use JET Server for Boot Params

By default, the OS provisioning server supplies the DHCP target host information for boot configuration. In cases of manual reboot for Solaris target hosts, you can use the JET server as an RARP boot params server instead.

Steps

1. **When you create the Generic Target host, set the `osp_control_service` variable to `FALSE`.**

   For information about creating the Generic Target host, see "Defining a Generic Target Host" on page 111. Setting the variable to FALSE changes the target host from an OS provisioning-served target host address to a JET-served target host address.

2. **To configure the JET server for RARP boot params service, change the `JS_CLIENT_MANAGEMENT` value in the `/opt/SUNWjet/etc/jumpstart.conf` file.**

   For example, on a SPARC system:

   ```
   JS_CLIENT_MANAGEMENT="bootp"
   ```

   Note – When booting the target host, an example console boot command to boot and install a target host over the network is `boot net - install`.

Using the `custom` Module

Use the `custom` module to install arbitrary packages, patches, and files on the target server. You can also use this module to run arbitrary scripts. When being installed on the target server, the order of installation is always: packages, patches, files, then scripts. If this order does not work for you, you can also create your own JET module. For information, see Appendix C.

The `custom` module can be used to install packages and patches at different stages of the build. See "Solaris Installation Process through JET" on page 54.

When you edit the target server template, you can list the names of the additional packages in the configuration variables `custom_packages`_[1-n], depending on when in the boot sequence the packages need to be installed. Likewise, you can identify patches in the variables custom_patches*_[1-n]*.

Packages and patches are installed in the order given, so you must order them appropriately to satisfy any dependencies between them. If you have many packages or patches, or if a common set is used frequently on several different target servers, consider creating your own module. For information, see Appendix C. You might also need to consider this approach if you need to intersperse the installation of packages with patches.

If the product installation includes packages to be installed, a `package.matrix` file is included which contains a list of supported operating systems and product version numbers with a list of the packages required to be installed. There is also a `patch.matrix` file which defines the required patches in a similar fashion.

Populating Custom Patch and Package Media

The `custom` module enables you to define a custom package and patch set on a per target server basis. This module also provides two scripts to enable the package and patch media to be placed in the correct place for the toolkit to find.

When copying patch and package media, the scripts will place the media according to the definitions of `JS_PKG_MEDIA` and `JS_PATCH_MEDIA` as found in the toolkit configuration file (`/opt/SUNWjet/etc/jumpstart.conf`). To use an alternative location to hold all the media, modify the configuration file before executing the scripts.

Custom patches can be placed using the `/opt/SUNWjet/bin/copy_custom_patches` script:

```
# copy_custom_patches src-dir patch [patch....]
```

This script takes at least two arguments, the first one is the source directory which contains the patches. Any other arguments are then considered to be patch numbers, which are located within the directory and subsequently copied.

Custom packages can be transferred using the `/opt/SUNWjet/bin/copy_custom_packages` script:

```
# copy_custom_packages src-dir arch package [package....]
```

The arguments are very similar to the arguments used for the custom scripts command, with the inclusion of the additional *arch* argument, which is used to define the target architecture for the packages. The target architecture will be defined by the output of `uname -p` on the target server. Currently, the values are either `sparc` or `i386`, for SPARC and IA86 architectures respectively.

Custom patches do not need the definition of an architecture, because distinctly numbered patches are released for each architecture for which the package is available.

The custom module cannot contain multiple different versions of the same package for the same architecture. If you need this functionality, consider a specific module to cover these requirements. See Appendix C.

Custom Files

The custom module enables files to be created, overwritten, or appended to on the target server. Files are referenced by a value that defines three items:

- The source file, relative to the target server directory within /opt/SUNWjet/Clients on the JumpStart server
- The mode of operation, which is either "a" (append) or "o" (overwrite)
- The destination file on the target server

For example, for a target server called banana the following line in the custom area of the template would append the contents of the file /opt/SUNWjet/Clients/banana/hosts on the JumpStart server to the file /etc/hosts on the machine banana while it was being built.

```
custom_files="hosts:a:/etc/hosts"
```

The source files must be placed correctly before the build of the target server. The files must be located within the /opt/SUNWjet directory tree.

Note – Do not use absolute paths for the source file.

Do not refer to files that are outside of the tree, such as /etc/hosts or /etc/passwd.

The middle field of the triple specifies whether to append or overwrite the destination file on the target server. If a set of files are common to a number of target servers, consider creating a holding area within the /opt/SUNWjet/Clients directory. Place the common files within that directory. The templates for the target servers can then refer to the files as:

```
custom_files="../common/hosts:a:/etc/hosts ../common/ftpusers:o:/etc/ftpusers"
```

Where the files are placed in /opt/SUNWjet/Clients/common rather than multiple copies in each target server specific directory. Because the template file is a Bourne shell script, you can use regular techniques to continue lines. Use the \ character, or append information to the variable custom_files="${custom_files} ......".

Custom Scripts

Custom scripts are defined by the source location of the script. The toolkit will copy the script to the scratch area on the target server during the build and execute it at the appropriate point.

As with the custom files described in the previous section, the source of the script must be within the `/opt/SUNWjet` directory. The source also can be a relative reference to a common holding area.

The custom module does not offer any provision for executing scripts before the first reboot, when the system is running on the NFS image from the JumpStart server and the real target server filesystem is mounted on `$ROOTDIR` (`/a`).

To execute a script prior to the first reboot, consider creating a module as described in Appendix C.

CHAPTER 6

Provisioning the Linux Operating Environment

This chapter explains how to use the OS provisioning plug-in to install the Linux operating system onto target hosts.

The chapter contains the following information:

Kickstart Technology

The OS provisioning plug-in uses Linux kickstart technology.

Linux Provisioning Process Overview

1. Set up the Linux image server.
2. Create the Linux image or attach an existing image to the image server.
3. Create a target host.
4. Install the OS onto the target host.

Setting up the Linux Server

The Linux boot and install server includes the following software:

- Linux (tested with RedHat Advanced Server 3.0)
- TFTP server
- NFS server
- N1 Grid SPS Remote Agent (RA) for Linux. For information, see "How to Prepare a Physical Host" in *N1 Grid Service Provisioning System 5.0 System Administration Guide*.

▼ How to Copy Linux Files

Steps

1. **Create a directory to contain the Linux files.**

   Note – To serve multiple distributions from the same server, create a separate directory for each distribution.

   ---

2. **Mount the first CD-ROM or ISO image file.**

   Use a command similar to one of the following examples:

   ```
   #mount /dev/cdrom          by default, will mount on /mnt/cdrom, check /etc/fstab
   #mount /path-to-first-iso /mnt/loop        assumes /mnt/loop exists
   ```

3. **Copy the content from the CD-ROM or ISO image file to the directory that you created in Step 1.**

   Use a command similar to one of the following examples:

   ```
   #/bin/cp -avf  /mnt/cdrom/* /export/RHEL3.0
   #bin/cp -avf  /mnt/loop/* /export/RHEL3.0
   ```

4. **Repeat Steps 2 and 3 for all the CD-ROM or ISO media in the distribution.**

▼ How to Set up NFS Access

The distribution should be accessible through NFS for each subnet served from this boot and install server.

Steps 1. **Edit the `/etc/exports` file and add lines similar to the following example:**

```
/export   10.42.42.*(ro)
```

2. **Enable NFS and start it.**

   Use commands similar to the following, which apply to RedHat AS 3.0:

```
#chkconfig nfs on
  #service nfs restart
```

▼ How to Set up TFTP Access

The distribution should be accessible through NFS for each subnet served from this boot and install server.

Steps 1. **Look for the *server_args* parameter in the `/etc/xinet.d/tftp` file.**

   This defines the root directory used by the TFTP server. If the directory does not exist, create it.

   Note – This task uses `/tftpboot` as an example.

   ---

2. **Enable TFTP and restart the xinetd daemon.**

   Use commands similar to the following example, which works on RedHat AS 3.0:

```
#chkconfig tftp on
  #service xinetd restart
```

3. **Copy or create the following directories under the TFTP root directory:**

   - `/tftpboot/pxelinux.0` – You can download the Linux PXE bootstrap file from `http://syslinux.zytor.com/pxe.php`.
   - `/tftpboot/pxelinux.cfg/` – This directory contains the PXE configuration files. These files contain parameters specific to each hardware platform and Linux distribution.
   - `/tftpboot/pxelinux.cfg/v20z-config` – A specific directory for the V20z platform.
   - `/tftpboot/vmlinuz` – The installation kernel found under the distribution, in `/images/pxeboot`.
   - `/tftpboot/initrd.img` – The installation RAM disk found under the distribution, in `/images/pxeboot`.

Example 6–1 Sample PXE Configuration File

The following example shows the PXE configuration file for a SunFire V20z system.

```
default el30 ks
serial 0 9600
label el30
kernel vmlinuz
append linksleep=30 ksdevice=eth0 console=tty0 console=ttyS1,9600
load_ramdisk=1 initrd=initrd.img network
```

Where:

- The first line (default) indicates to use label el30 and do a Kickstart installation (passing ks to the kernel).
- The third line (label) defines a label el30.
- The fourth line (kernel) identifies the kernel to load. The path is relative to pxelinux.0.
- The fifth line (append) contains a series of network configuration parameters. The append parameters should be all on the same line. The specific parameters shown here are as follows:

  linksleep=30 tells the kernel to wait before determining that the system is not working. Some network interfaces require a long time to come up.

  ksdevice=eth0 indicates to perform the installation through the eth0 interface. If this parameter us omitted, eth0 is assumed.

  console=tty0 console=ttyS1,9600 specify console parameters to show the installation output on the console. The parameters vary depending on the target hardware platform.
- The last line (load_ramdisk and initrd) indicate to use initrd.img for the RAM disk and to perform a network installation. The path for the RAM disk is relative to pxelinux.0

Note – vmlinuz and initrd files only work with a specific version of a specific distribution. If using the server to provide multiple distributions, or multiple versions of a distribution, each vmlinuz and initrd pair must be uniquely identifiable. You can create the unique identifiers either by using separate directories or by renaming the files, as shown in the following example:

```
/tftpboot/pxelinux.0
/tftpboot/RHEL2.1/vmlinuz
/tftpboot/RHEL2.1/initrd
/tftpboot/RHEL3.0/vmlinuz
/tftpboot/RHEL3.0/initrd

or

/tftpboot/pxelinux.0
/tftpboot/rhel2.1-vmlinuz
/tftpboot/rhel2.1-initrd
/tftpboot/rhel3.0-vmlinuz
/tftpboot/rhel3.0-initrd
```

▼ How to Create the Linux Image Server

You can create a Linux image server from the N1 Grid SPS browser interface or command-line interface. To create the server from the command line, type commands similar to the following:

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/LinuxServer \
-name "linuxserver" -u admin -p admin -vars "boot_server_tftp_root_directory=/tftpboot"
# cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/untyped/LinuxServer-create \
-tar H:NM:bikickstart -comp + -vs linuxserver -pto 30 -nto 10
```

To create the server from the browser interface, follow these steps.

Before You Begin The kickstart file should be placed on the Linux boot and install server in an NFS exported directory so that the target hosts can locate it, for example, /export/kickstart.

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the Linux Image Servers section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the LinuxServer component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the LinuxServer component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **If needed, change the boot_server_tftp_root_directory variable to the location where you want the Linux packages to be installed.**

     By default, the path is /tftpboot.

     d. **If needed, change the linuxHost variable to the location of the command-line interface.**

     By default, the virtual Linux host name is set to the host name of the target host with -linux appended.

Caution – Do not modify the `installPath` variable.

e. **Save the variables set.**

f. **Select the variables set that you just saved from the drop-down menu in the LinuxServer component row of the Plan Parameters table.**

5. **Select the host on which to create the Linux image server.**

6. **Click Run Plan (includes preflight).**

7. **To verify that the image server is created successfully, click the Hosts link in the left side of the provisioning server window.**

   You should see the name of the virtual host appended with `-linux`. For example, if you installed on host `bikickstart`, the virtual host is `bikickstart-linux`.

Creating Linux Images and Profiles

▼ How to Create a Linux Provisioning Profile (Browser Interface)

Before You Begin Make sure the Linux image server exists and that the Linux operating system software is available to that server. For more information about creating the Linux image server, see "Setting up the Linux Server" on page 70.

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create Profile in the Linux Images section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the LinuxImage component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the LinuxImage component row of the Plan Parameters table.**

a. **Click Create Set.**

b. **Type a name for the variables set.**

c. **Verify variables and change them, if needed.**

d. **Save the variables set.**

e. **Select the variables set that you just saved from the drop-down menu in the LinuxImage component row of the Plan Parameters table.**

5. **On the Plan Details Run page, select the Linux image server on which to create the profile.**

Tip – The host name of the Linux image server ends in `-linux`.

---

6. **Click Run Plan (includes preflight).**

7. **To verify that the Linux profile is created to the Linux image server, click Components in the left pane of the N1 Grid SPS browser interface.**

   You should find a `provision` component in the folder `/com/sun/n1osp/autogen-`*server-name*`-linux/provision/`, where *server-name* is the Linux server name that you provided in the variable set. For example: `/com/sun/n1osp/autogen-LinuxRedHat3.0_x86-linux/provision/`.

▼ How to Create a Linux Provisioning Profile (Command-Line Interface)

Steps

1. **To create a variable set for the `LinuxImage` component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/LinuxImage \
-name "redhat3" -u admin -p admin -vars "osp_server=masterserver-osp;version=RedHat3.0; \
architecture=x86;installPath=Linux:[version]_:[architecture]; \
description=testlinux;min_disk_size=10GB;monitor_install=FALSE; \
boot_server_name=bikickstart;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.3;kickstart_file=/export/el30.cfg;\
pxe_boot_file=linux-install/pxelinux.0;pxe_configuration_file=el30-console"
```

2. **To run the plan, type a command similar to the following example:**

```
# cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/untyped/LinuxImage-create-profile \
-tar H:NM:bikickstart-linux -comp + -vs redhat3 -pto 30 -nto 10
```

Installing the OS on the Target Host

Profiles are used to provision OS Images to target Hosts. These Profiles are found in `/com/sun/n1osp/autogen...provision` folders. One new Profile is added to a provision folder each time a new OS Image is created (or attached).

▼ How to Install the OS on the Target Host

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**
2. **On the OS Provisioning Common Tasks page, click Provision OS in the Linux Images section.**
3. **Navigate to the Linux folder that contains the `provision` component.**

   For example,
   `/com/sun/n1osp/autogen-bikickstart-linux/provision/LinuxRedHat3.0_x86`
4. **Slect the component name.**
5. **On the Component Details page, click the Run action next to the Provision_start:Install procedure.**
6. **Select the target host on which to provision the operating system.**

   Tip – The target host should have a host name that ends in `-target`.

   ---
7. **Click Run Plan (includes preflight).**

▼ How to Check OS Provisioning Status

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**
2. **On the OS Provisioning Common Tasks page, click Status in the OS Provisioning Administration Tasks section.**
3. **On the Plans Details page, click Run.**

4. **On the Plan Details Run page, select the target host on which you provisioned the operating system.**

5. **Click Run Plan (includes preflight).**

6. **Follow the Details links to view the status.**

CHAPTER 7

Provisioning the Windows Operating Environment

This chapter explains how to use the OS provisioning plug-in to install the Windows operating system onto target hosts.

The chapter contains the following information:

Note – Although you can provision Windows 2000 operating systems, the Windows boot and install server must be running the Windows 2003 operating system.

Remote Installation Services (RIS) Technology

The OS provisioning plug-in uses RIS technology to provision Windows systems. The Remote Installation Services provide an independent setup routine that is executed on the RIS server. RIS provides a centralized location to integrate maintenance and troubleshooting tools that are accessible through a network boot. RIS provides network administrators with the capability to easily install the base operating system or to replace a system that has failed. For more information about RIS, see the Microsoft documentation.

Windows Provisioning Process Overview

1. Set up the Windows image server.
2. Create the Windows image or attach an existing image to the image server.
3. Create a target host.
4. Install the image onto the target host.

Setting Up the Windows Server

Setting up the Windows boot and install server consists of several tasks:

1. Set up Windows Active Directory and RIS services on the Windows RIS server.
2. Install OS provisioning scripts on the Windows RIS server.
3. Install Windows Remote Shell Service (RSH) or Secure Shell Service (SSH) on Windows RIS server.
4. Set up a Windows RIS image on the Windows RIS server.
5. Configure the Setup Information (`.sif`) file for the above image.
6. Configure the OSchooser files for OS provisioning image deployment.

▼ How to Set Up the Windows RIS Server

Steps

1. **Install Windows 2003 Server Edition operating system on the Windows boot and install server.**

   Create a separate NTFS partition that is big enough to store at least one image of roughly 700 Mbytes to hold RIS images. You will need to identify this partition when you make the server into a RIS server.

2. **Install and configure Active Directory Server on this machine.**

   Identify this system as a domain controller. For information about Active Directory Server, see Microsoft Knowledge Base Article 324753.

3. **Install and configure RIS server on this machine.**

   For more information, see Microsoft Knowledge Base Article 325862.

4. **Provide appropriate RIS server settings.**

In the RemoteInstall properties for the RIS server, set the following two variables:

- Respond to client computers requesting service
- Do not respond to unknown client computers

5. **Add the RIS server's IP address (Provisioning subnet interface IP address) in the DHCP snap-in:- "Manage Authorized Servers" list.**

6. **Disable any kind of DHCP service if its already running on the RIS server.**

7. **Configure user accounts for the RIS server.**

   Create two user accounts in the RIS server's Active Directory and add them to the groups Administrators, Domain Admins, and Domain Users:

   - OS provisioning administration user – For example, `username n1ospadmin`.
   - OS provisioning RSH/SSH user – The RSH user name should always be root because the DHCP service on the OS provisioning server runs as root. The SSH user name can be any valid username.

8. **Run the "Delegate control" wizard for the domain controller in the Active Directory Users and Computers snap-in:**

   a. **Select Join a Computer to the Domain.**

   b. **Set the user to the username that you defined in the previous step for the OS provisioning administration user.**

▼ How to Install OS Provisioning Scripts on the Windows RIS Server

Steps

1. **Copy the following Windows OS provisioning scripts, sample files, and tools from the product media to a user-specified directory:**

   - */dvd/release/n1osp/1.0/Windows/*`SampleOSCFiles`
   - */dvd/release/n1osp/1.0/Windows/*`SampleSIFFiles`
   - */dvd/release/n1osp/1.0/Windows/*`IspScripts`
   - */dvd/release/n1osp/1.0/Windows/*`IspTools`

   Where */dvd/release/n1osp/1.0/Windows/* is the path from the product media to the Windows directory that contains these files.

2. **Create a folder to store the customized SIF files that are generated during provisioning.**

   Put this folder under the RemInst share (for example, `D:\RemoteInstall\`*folder-name*). Name the folder something relevant, like `n1osp`.

3. **Give read and write access to the folder that you created in the previous step for the OS provisioning administration user and the RSH/SSH user.**

4. **Verify that the OS provisioning administration user and the RSH/SSH user have read, write, and execute permissions to the temporary folder in the RemInst share.**

   For example, `D:\RemoteInstall\tmp`. This access is needed so that the scripts can remove any stale temporary SIF files created by BINLSVC as a result of previous provisioning operations.

▼ How to Install Windows RSH Server on the Windows RIS Server

The OS provisioning plug-in supports WinRsh (Native Windows RSH Service) for Windows 2003.

Steps

1. **Install Windows 2000/2003 Resource Kit and Windows Services for Unix 3.5 (SFU3.5) software on the Windows boot and install server.**

   You can install this software either from Windows Resource Kit CD or an online location. These packages include Windows Rsh service and other tools like `setupmgr`, `deploy.cab`, and `xcacls` used by the OS provisioning scripts. For SFU install instructions, see Microsoft Knowledge Base Article 321712 .

2. **Install and configure Windows Remote Shell Service (WinRsh service) on the RIS server as N1 RSH user (that is, as user root).**

   For more information, see `http://www.microsoft.com/technet/interopmigration/unix/sfu/sfu35rsh.mspx`.

   Note – Incorrect file permissions can cause WinRsh to fail. Symptoms of this problem include an Access denied message to the client and a log entry in the Event Viewer. To resolve this problem, assign full permissions to the user group Administrators/SYSTEM in the `.rhosts` ACL list.

   The following example shows the contents of the *system-drive*`/windows/system32/drivers/etc/.rhosts` file.

   ```
   ------------------
   10.5.133.22 root
   ------------------
   ```

▼ How to Install Windows SSH Server on the Windows RIS Server

The OS provisioning plug-in supports Cygwin's OpenSSH package for Windows 2003.

Step ● **Install and configure Cygwin's Open Secure Shell Service (OpenSsh service) for Windows 2003 on the RIS server as N1 SSH user.**

For more information, see the Cygwin Home Page and the Open Ssh Windows Install Instructions.

▼ How to Set Up a Windows Image on the Windows RIS Server

Steps

1. **Copy the Windows distribution from a CD or Network Share to the RIS server CIFS.**

   Use the `RiSetup.exe` utility.

   Note – Do not use MSDN Combo CDs to create a RIS Image. You will get an error message that was described in the Microsoft Knowledge Base Article #300556. To resolve this issue, please use a retail version of the OS image being created.

   ---

2. **Add the required drivers to the RIS image created in the previous step.**

   Be sure to include all the drivers, such as those for the network adapter, SCSI hard drives, video, and modem.

   For more information about adding drivers to RIS images, see Microsoft Knowledge Base Article 315279, Microsoft Knowledge Base Article 246184, Microsoft Knowledge Base Article 254078, and information about adding mass storage devices to RIS.

   Note – Create the distribution folder accordingly.

   See the sample setup files in `SampleSIFFiles` folder for more details.

3. **Create the folder** *product-directory*`/$OEM$/$1/N1ISP/` **under the image distribution folder.**

   product-directory is the Windows image distribution folder that has "i386" as a subfolder that contains the kernel files.

   For example:

   ```
   D:\RemoteInstall\Setup\English\Images\WinAS_2000\$OEM$\$1\N1ISP
   D:\RemoteInstall\Setup\English\Images\WinAS_2000\i386
   ```

4. **Copy the `assignIPInfo.exe` file from the Windows boot and install server `IspTools` folder to the** *product-directory*`/$OEM$/$1/N1ISP/` **folder that you created in the previous step.**

Note – The script expects the N1ISP folder to be part of the image. Copy the tools required for provisioning to this directory path *product-directory*/$OEM$/$1/N1ISP/. Do not change the N1ISP folder name.

Tip – You created the Windows boot and install server IspTools folder in step 1 of "How to Install OS Provisioning Scripts on the Windows RIS Server" on page 81.

You can also copy the N1 Grid SPS RA binary to the above folder, if you want to install a RA on the target machine.

The following example shows a sample directory structure.

```
D:\RemoteInstall\Setup\English\Images\WIN2003_TRIAL\$OEM$\$1\N1ISP>dir
  Volume in drive D is Local Disk
  Volume Serial Number is A8D9-4012

Directory of D:\RemoteInstall\Setup\English\Images\WIN2003_TRIAL\$OEM$\$1\N1ISP

  02/04/2005  02:39 PM    <DIR>          .
  02/04/2005  02:39 PM    <DIR>          ..
  02/04/2005  11:34 AM            29,184 assignIPInfo.exe
  01/31/2005  03:33 PM        27,595,776 cr_ra_win32_5.0.msi
                 2 File(s)     27,624,960 bytes
                 2 Dir(s)  23,775,776,768 bytes free
```

▼ How to Configure the Setup Information for the Windows Image

Setup information for Windows is stored in a RIS configuration profile. See the sample setup files from SampleSIFFiles folder for more details.

Steps

1. **Edit the configuration profile file for each distribution to set the parameters mentioned in the following steps.**

   To edit this file, use the SetupMgr.exe utility. from the Windows Resource Kit CD (deploy.cab file). You can also manually edit the file using the notepad tool.

   For a Windows 2003 image, the default SIF file is on the RIS Server at the following location:

   ```
   //Reminst/Setup/English/Images/Windows2003/i386/templates/ristndrd.sif
   ```

2. **Verify or add the following configuration parameters to the `[data]` section in your `ristndrd.sif` configuration file:**

```
AutoPartition=1
UnattendedInstall="Yes"
```

3. **Verify or add the following configuration parameters to the `[Unattended]` section in your `ristndrd.sif` configuration file:**

```
UnattendMode=FullUnattended
NtUpgrade=No
OverwriteOemFilesOnUpgrade=No
OemPreinstall = Yes
```

4. **Verify or add the following configuration parameters to the `[UserData]` section in your `ristndrd.sif` configuration file:**

```
FullName = "N1isp SunUser"
OrgName = "Sun Microsystems"
ProductKey=XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Note – Replace the `FullName` and `OrgName` values with something that is appropriate for your environment.

While deploying some operating systems, such as Windows 2000 Advanced Server, `ProductKey` does not work. Instead, use a `ProductID` variable.

---

5. **Verify or add the following configuration parameters to the `[GuiUnattended]` section in your `ristndrd.sif` configuration file:**

```
AdminPassword="sun123"
AutoLogon=Yes
AutoLogonCount=1
EncryptedAdminPassword=NO
```

Note – `AutoLogonCount=1` implies that the super-user administrator is automatically logged on to the target host after installation with the default password as `sun123` for the very first time. This is required to execute the provision time commands specified in the [GuiRunOnce] section in the SIF file. When the Administrator user attempts to login again, he will be prompted to change the password.

---

6. **Verify or add the following sections to your `ristndrd.sif` configuration file:**

```
[Branding]
        BrandIEUsingUnattended=Yes
[Proxy]
```

```
Proxy_Enable=0
Use_Same_Proxy=0
```

Tip – The `SetupMgr.exe` utility generally adds these sections for you.

---

7. **Add any specific plug and play drivers to the image.**

   See the instructions in Microsoft Knowledge Base article #: 315279.

▼ How to Configure OS Chooser Files for Image Deployment

The following procedure explains how to configure the OS chooser files for English users. The procedure might be slightly different for other languages. For more information, see the Microsoft documentation.

Steps

1. **Copy the following files in the `//RemoteInstall/OSChooser/i386` directory on the RIS server.**

   a. **Create a backup copy of the `startrom.com` file to `startrom.bak`.**

   b. **Copy the `startrom.n12` file to `startrom.com`.**

2. **Edit the CIW Screens to simplify user interactions.**

   The following sub-steps explain how to make changes that simply the following features:

   - Add Autoenter functionality for all the required `.osc` screens. This capability eliminates all user interactions with CIW screens during the installation phase.
   - Add user information for authentication in the `login.osc` file. This step eliminates the user interaction during installation. Automating the user information in the `login.osc` file authorizes and enables the target host to download and install files directly from the RIS server.
   - Add an OS provisioning customized path to the image in the `oschoice.osc` file. This step points the oschoice CIW screen to a known RIS image location for the `.sif` file. This value points to a customized path to an OS provisioning temporary `.sif` file that is generated during provisioning run-time.

   a. **Create backup copies of the following files:**

      - `//RemoteInstall/OSChooser/Welcome.osc`
      - `//RemoteInstall/OSChooser/English/install.osc`
      - `//RemoteInstall/OSChooser/English/login.osc`
      - `//RemoteInstall/OSChooser/English/oschoice.osc`

- `//RemoteInstall/OSChooser/English/warning.osc`

  For example:

```
copy //RemoteInstall/OSChooser/Welcome.osc
     //RemoteInstall/OSChooser/Welcome.osc.bak
copy //RemoteInstall/OSChooser/English/install.osc
     //RemoteInstall/OSChooser/English/install.osc.bak
copy //RemoteInstall/OSChooser/English/login.osc
     //RemoteInstall/OSChooser/English/login.osc.bak
copy //RemoteInstall/OSChooser/English/oschoice.osc
     //RemoteInstall/OSChooser/English/oschoice.osc.bak
copy //RemoteInstall/OSChooser/English/warning.osc
     //RemoteInstall/OSChooser/English/warning.osc.bak
```

b. **Copy the supplied `Welcome.osc` file to `//RemoteInstall/OSChooser`.**

c. **Copy the supplied `install.osc`, `login.osc`, `oschoice.osc`, and `warning.osc` files from the `SampleOSCFiles` folder to `//RemoteInstall/OSChooser/English`.**

d. **Modify the INPUT tags for `USERNAME` and `PASSWORD` in the `login.osc` to match the OS provisioning administration user name and password.**

```
<FORM ACTION="CHOICE">
    &nbsp&nbspUser name: <INPUT NAME="USERNAME" MAXLENGTH=255 TYPE=TEXT VALUE=n1ispadmin>
    &nbsp&nbsp&nbspPassword: <INPUT NAME="*PASSWORD" TYPE=PASSWORD MAXLENGTH=20 VALUE=sun123>
    <BR>
      ...
      ...
      ...
</FORM>
```

Tip – You created the administration user and password in step 7 of "How to Set Up the Windows RIS Server" on page 80.

Note – Do not modify any other information.

e. **Modify the OPTION tag in the `oschoice.osc` file with the path that points to the location of the OS provisioning temporary `.sif` file that is generated during provisioning runtime.**

Generally, this file path contains the folder name that you created in step 3 of "How to Install OS Provisioning Scripts on the Windows RIS Server" on page 81, appended with the GUID file name that is created at provisioning time as `%guid%.sif`. For example, `D:\RemoteInstall\N1Isp`*guid*`.sif`.

```
<OPTION VALUE="D:\RemoteInstall\N1Isp\%guid%.sif" TIP="This will install
N1 ISP customized Windows OS Image" SELECTED>%guid%</SELECT>
```

▼ How to Set BIOS Boot Order on the Target Host

Steps

1. **Set the BIOS Settings on the target host in the following order:**

   a. **Boot from Network (PXE-enabled)**

   b. **Boot from Hard Disk**

   c. **Boot from CD**

   d. **Boot from Floppy**

2. **Identify the target host's primary NIC MAC address for a PXE network boot and the GUID value.**

   You can obtain the GUID from the target host's BIOS settings or from any of the following other sources:

   - A label on the side of the computer case.
   - A label within the computer case.
   - The basic input/output system (BIOS) of the client computer.
   - Network traffic from the client computer. You can use a network utility to sniff the network traffic and locate the DHCPDiscover packet. That field will contain the 128-bit, 16-byte GUID/UUID.

Note – The GUID must be in the form {*dddddddd-dddd-dddd-dddd-dddddddddddd*}, where *d* is a hexadecimal text digit. For example, `{921FB974-ED42-11BE-BACD-00AA0057B223}`. Valid entries for the client GUID are restricted to the numbers 0 through 9, lowercase alphabetical characters, uppercase alphabetical characters, and a dash. Spaces, underscores, and other special symbols are not permitted in a GUID.

▼ How to Create the Windows Image Server (Browser Interface)

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the Windows Image Servers section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

- **To use an existing variables set, select a name from the drop-down menu in the WindowsServer component row of the Plan Parameters table.**
- **To create a new variables set, click Select from List in the WindowsServer component row of the Plan Parameters table.**

  a. **Click Create Set.**

  b. **Type a name for the variables set.**

  c. **Verify variables and change them, if needed.**

     The image server component has only a few variables, most of which you are unlikely to change. For a list of those variables and their default values, see "Windows Image Server Component Variables" on page 90.

  d. **Save the variables set.**

  e. **Select the variables set that you just saved from the drop-down menu in the WindowsServer component row of the Plan Parameters table.**

5. **Select the host on which to create the Windows image server.**

6. **Click Run Plan (includes preflight).**

7. **To verify that the image server is created successfully, click the Hosts link in the left side of the provisioning server window.**

   You should see the name of the virtual host appended with `-windows`. For example, if you installed on host `masterserver`, the virtual host is `masterserver-windows`.

▼ How to Create the Windows Image Server (Command-Line Interface)

To create the Windows image server from the command line, follow these steps.

Steps

1. **Create a variables set.**

   Use a command similar to the following example:

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/WindowsServer -name "winRisServer"  \
-u admin -p admin -vars "active_directory_userid=n1ispadmin; \
active_directory_password=Clz6pK2b6qw=;boot_server_access_protocol=rsh; \
boot_server_access_userid=root;boot_server_access_password=; \
ciw_directory=D:\\\\RemoteInstall\\\\OSChooser\\\\English; \
temporary_sif_directory=D:\\\\RemoteInstall\\\\N1isp;installPath=C:\\\\N1ispRisProj"
```

2. **Run the plan.**

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/n1osp/untyped/WindowsServer-create \
-tar H:NM:masterserver -comp + -vs winRisServer -pto 30 -nto 10
```

Windows Image Server Component Variables

TABLE 7–1 Variables for Windows Image Server Component

Variable Name	Description	Default Value
`installPath`	Location of OS provisioning scripts	`D:\n1isp\scripts`
`active_directory_userid`	User ID for the active directory	`n1risroot`
`active_directory_password`	Encrypted password for the user defined above. For information about encrypting the passwords, see "Password Encryption" on page 113.	
`boot_server_access_protocol`	Protocol to access boot server	`rsh`
`boot_server_access_userid`	User ID for the boot server	`root`
`boot_server_access_password`	Encrypted password for the user defined above. For information about encrypting the passwords, see "Password Encryption" on page 113.	
`ciw_directory`	Directory of CIW/osc files e.g., `D:\RemoteInstall\OSChooser\English`	
`temporary_sif_directory`	Location of temporary sif files in the RemInst share e.g., `D:\RemoteInstall\tmp`	
`windowsHost`	Virtual host representing the Windows Server	`:[target:sys.hostName]-windows`

Creating Windows Images and Profiles

A provisioning profile identifies the OS files to use and includes additional information about how those files are to be deployed.

▼ How to Create a Windows Provisioning Profile (Browser Interface)

Before You Begin Make sure the Windows image server exists and that the Windows OS software is available to that server. For more information about creating the Windows image server, see "Setting Up the Windows Server" on page 80.

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create Profile in the Windows Images section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the WindowsImage component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the WindowsImage component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Verify variables and change them, if needed.**

        If you need more information than is provided in the prompts in the component variables table, see the detailed list of components, along with default values and examples, in "Windows Provisioning Profile Variables" on page 92.

     d. **Save the variables set.**

     e. **Select the variables set that you just saved from the drop-down menu in the WindowsImage component row of the Plan Parameters table.**

5. **On the Plan Details Run page, select the Windows image server on which to attach the image.**

   Tip – The host name of the Windows image server ends in `-windows`.

   ---

6. **Click Run Plan (includes preflight).**

7. **To verify that the Windows image is attached to the Windows image server, click Components in the left pane of the N1 Grid SPS browser interface.**

   You should see a component of type `com.sun.n1osp#Provision` that has a description similar to `Windows provisioning profile`.

▼ How to Create a Windows Provisioning Profile (Command-Line Interface)

Steps

1. **To create a variable set for the `WindowsProfile` component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/LinuxImage \
-name "redhat3" -u admin -p admin -vars "osp_server=masterserver-osp;version=RedHat3.0; \
architecture=x86;installPath=Linux:[version]_:[architecture];description=testlinux; \
min_disk_size=10GB;monitor_install=FALSE;boot_server_name=bikickstart; \
boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.3;kickstart_file=/export/el30.cfg; \
pxe_boot_file=linux-install/pxelinux.0;pxe_configuration_file=el30-console"
```

   For more information about the variables, see "Windows Provisioning Profile Variables" on page 92.

2. **To run the plan, type a command similar to the following example:**

```
 # cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/n1osp/untyped/LinuxImage-attach \
-tar H:NM:bikickstart-linux -comp + -vs redhat3 -pto 30 -nto 10
```

Windows Provisioning Profile Variables

The following table lists all component variables for the Windows provisioning profile. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE 7–2 Component Variables for Windows Provisioning Profile

Variable Name	Description	Default or Sample Value
`osp_server`	Virtual server that represents the OS provisioning service	`masterserver-osp` (Example)
`version`	Windows version	`2003` (Default)
`architecture`		`x86` (Default)
`installPath`	Name of Windows image	`Windows:[version]_:[architecture]` (Default)
`description`	Description	
`min_disk_size`	Minimum disk size for this profile to be deployed	`10GB` (Default)
`monitor_install`	If installation has to be monitored	`False` (Default)
`boot_server_name`	Name of Windows boot server	

TABLE 7–2 Component Variables for Windows Provisioning Profile *(Continued)*

Variable Name	Description	Default or Sample Value
`boot_server_subnet_address`	Subnet served by boot server	
`boot_server_subnet_ip_address`	IP address of boot server on this subnet	
`boot_srv_active_directory_domain`	Windows active directory domain for this subnet	`winprov.n1lab.sun.com` (Example)
`windows_product_directory`	Windows product directory name on boot server	`D:\RemoteInstall\Setup\English\Images\Win2003` (Example)
`sif_file`	Location of software information file (SIF)	`D:\RemoteInstall\Setup\English\Images\Win2003\i386\templates\ristndrd.sif` (Example)
`windows_component_folder`	Location of auto-generated Windows Provision component folder	`/com/sun/n1osp/autogen-:[target(..):sys.hostName]-windows` (Default)

Installing the OS on the Target Host

Profiles are used to provision OS Images to target Hosts. These Profiles are found in `/com/sun/n1osp/autogen...provision` folders. One new Profile is added to a provision folder each time a new OS Image is created (or attached).

▼ How to Install the OS on the Target Host

Steps

1. **In the Common Tasks section of the provisioning software page, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Provision OS in the Windows Images section.**

3. **Navigate to the Windows folder that contains the `provision` component.**

   For example, `/com/sun/n1osp/autogen-masterserver-windows/provision/`.

4. **Click the component name.**

   For example, `Windows2000_x86`

5. **On the Component Details page, click the Run action next to the Provision_start:Install procedure.**

6. **Select the target host on which to provision the operating system.**

   Tip – The target host should have a host name that ends in `-target`.

   ---

7. **Click Run Plan (includes preflight).**

CHAPTER 8

Target Hosts for OS Provisioning

When you provision the operating system, you have to know something about the hardware that you are provisioning. To simplify this process, the OS provisioning plug-in provides a set of target host definitions. The OS provisioning plug-in uses this information to remotely manage the target servers and perform functions such as power off, power on, getting console capability, and setting boot mode of the target.

This chapter explains the types of target hosts that you can easily provision and provides several detailed examples. The chapter contains the following information:

Target Host Types

Different mechanisms are used to remotely manage different hardware types. As a result, you need to know some specific information before you can effectively configure the operating system for that kind of hardware. The OS provisioning plug-in provides the following hardware groups for you to use for your OS provisioning targets:

- Sun Intelligent Platform Management Interface (IPMI) Targets
- Sun Advanced Light Out Management (ALOM) Targets: View Create Manage Sun based hosts
- Sun Light Out Management (LOM) Targets
- Sun Mid Range Domain Controller Targets
- Hewlett-Packard Integrated Light Out Management (ILO) Targets
- Other IPMI Targets

- Other Remotely Managed Targets (not belonging to any of the above categories, such as B100s and B100x)
- Generic target hosts with no remote management support

Each target host type has a specific set of variables that apply to that host type. For detailed information about the variables that apply to each target host type, see Appendix A.

Target Host Types by Platform

The following table lists some of the platforms and targets that the OS provisioning plug-in supports. For each hardware platform, the table shows the following information:

- The firmware or service processor versions that have been tested
- The configuration that needs to be performed on these platforms
- The information that needs to be gathered for the OS provisioning variables
- The target host type that applies

TABLE 8–1 Target Hosts by Platform

Target Platform	Firmware or Service Processor	Configuration Requirements	Information Needed for OS Provisioning Target Variables	Target Host Type
SunFire v240, v210 family	Sun Advanced Lights Out Manager 1.3 (ALOM)	■ Provide IP (static) to ALOM port (routable from OS provisioning server) ■ Enable telnet	■ ALOM ip, user/password ■ MAC address of the target provisioning interface	Sun ALOM target

TABLE 8–1 Target Hosts by Platform *(Continued)*

Target Platform	Firmware or Service Processor	Configuration Requirements	Information Needed for OS Provisioning Target Variables	Target Host Type

SunFire v120 family	Sun Lights Out Manager version: v3.12 (LOM)	■ Connect LOM port to terminal server (Terminal server IP should be routable from OS provisioning server)
		■ To share Serial A/LOM between the LOM and the Console, setup console connections to both the Serial A/LOM and the Serial B port For example, from the Solaris command line:
		`eeprom input-device=ttya`
		`eeprom output-device=ttya`
		`reboot`
			■ LOM user/password (if any)
			■ MAC address of the target provisioning interface
			■ Terminal server information (type, ip, port connected to Target LOM port)
				Sun LOM target

SunFire v20z, v40z family	Sun IPMI v1.5 Service Processor, Version V2.1.0.10	■ Provide IP (static) to Network Management port (routable from OS provisioning server) ■ Enable ssh ■ Enable ipmi, ipmi channel “lan” ■ Configure BIOS to boot from PXE-enabled NICs first	■ Service Processor IP ■ ssh user/password ■ ipmi password ■ MAC address of the target provisioning interface ■ GUID of the provisioning interface	Sun IPMI target

TABLE 8–1 Target Hosts by Platform *(Continued)*

Target Platform	Firmware or Service Processor	Configuration Requirements	Information Needed for OS Provisioning Target Variables	Target Host Type
SunFire B100s (sparc), B100x (x86) family	Sun Advanced Lights Out Manager for Blade Servers 1.2 (ALOM-B 1.2)	■ Provide IP (static) to ALOM port (routable from OS provisioning server) or connect ALOM serial port to terminal server (Terminal server IP should be routable from OS provisioning server) ■ Enable Telnet (for IP-based managment)	■ ALOM IP, or terminal server information (ip, type, port connected to ALOM serial port) ■ ALOM user/password ■ MAC address of target provisioning interface ■ GUID of the provisioning interface (for B100x only)	Other Remotely Managed Target using `b1600sc.jar` as the Network Management HAL jar power, boot and console services are all supported by this HAL
SunFire v60x, v65x family	IPMI-enabled Baseboard Management Controller (BMC)	■ Provide IP (static) to LOM/IPMI port (routable from OS provisioning server) or enable ipmi, ipmi channel "lan" ■ Configure BIOS to boot from PXE-enabled NICs first	■ LOM/IPMI IP ipmi password ■ MAC address of the target provisioning interface ■ GUID of the provisioning interface	Other IPMI target using `ipmi.jar` as the Network Management HAL jar only power, service is supported by this HAL
SunFire 3800, 4800 or 6800 domain	System Controller: ScApp version: 5.18.1 Build_01 RTOS version: 41	■ Provide IP (static) to System Controller (routable from OS provisioning server) ■ Enable telnet	■ Platform SC password ■ Domain SC password ■ MAC address of the target provisioning interface	Sun Mid Range Target

TABLE 8–1 Target Hosts by Platform *(Continued)*

Target Platform	Firmware or Service Processor	Configuration Requirements	Information Needed for OS Provisioning Target Variables	Target Host Type
HP Proliant DL Series (supporting iLO-based managment)	iLO Integrated Lights Out Manager v 1.6.41	■ Provide IP (static) to iLO port (routable from OS provisioning server) ■ Configure BIOS to boot from PXE-enabled NICs first	■ iLO ip, user/password ■ MAC address of the target provisioning interface ■ GUID of the provisioning interface	HP iLO Target
Other Targets	-	Configure BIOS to boot from PXE-enabled NICs first (x86 only)	■ MAC address of the target provisioning interface ■ GUID of the provisioning interface (x86 only)	Generic Target (manual control)

Example Tasks for Defining Target Hosts

The general process that you follow to define a target host is similar for all target host types. However, the details differ for each type. This section contains sample tasks for some, but not all, target host types.

Defining a Sun ALOM Target Host

Before you can install the image onto a system, often referred to as *provisioning*, you must inform the OS provisioning server about that system.

▼ How to Create a Sun ALOM Target Host (Browser Interface)

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the Sun ALOM section of the Target Hosts section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the SunALOM component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the SunALOM component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Verify variable values and change them, if needed.**

        If you need more information that is provided by the prompts in the component variables table, see the detailed list of variables, along with default values and examples, in "Sun ALOM Target Host Variables" on page 116.

     d. **Decide whether to enter passwords in the variables set or in the plan prompts.**

        There are two methods for entering the passwords:

        - Type appropriate values in the following variables:

          ```
          alom_access_userid
          alom_access_password
          terminal_server_userid
          terminal_server_password
          ```

          For information about encrypting the passwords, see "Password Encryption" on page 113.

        - Leave the component variables blank and enter the password in the plan prompts provided in the Plan Details Run page.

     e. **Save the variables set.**

     f. **Select the variables set that you just saved from the drop-down menu in the SunALOM component row of the Plan Parameters table.**

5. **From the Plan Details Run page, select the provisioning server on which to define the target host.**

Tip – The host name of the OS provisioning server ends in `-osp`.

6. **To encrypt the passwords from the plan, follow these steps on the Plan Details Run page.**

   a. **Click the checkbox next to "If you are specifying the ALOM password below."**

   b. **Type the password in the Password to Access ALOM field.**

      The password encrypts as you type.

   c. **Click the checkbox next to "If you are specifying the terminal server password below."**

   d. **Type the terminal server password in the Terminal Server Password field.**

      The password encrypts as you type.

7. **Click Run Plan (includes preflight).**

8. **To verify that the target host is created successfully, click the Hosts link in the left side of the N1 Grid SPS browser interface.**

   You should see the name of the virtual host appended with `-target`. For example, if you defined a provisioning target for host `test240`, the virtual host is `test240-target`.

▼ How to Create a Sun ALOM Target Host (Command-Line Interface)

Steps

1. **To create a variable set for the SunALOM component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/targets/SunALOM  \
-name "testv240" -u admin -p admin -vars "installPath=testv240; \
description=To test v240;ethernet_mac_address=0:3:ba:53:5b:5d; \
ethernet_ip_address=10.42.42.100;ethernet_netmask=255.255.255.0;architecture=sun4u; \
kernel_arch=sun4u;disk_size=20GB;alom_ip_address=10.5.133.150;alom_access_userid=admin; \
alom_access_password="
```

   For more information about the variables, see "Sun ALOM Target Host Variables" on page 116.

2. **To set up the appropriate password access, create a file that contains the following entries:**

```
true
root
false
```

```
root
```

Where:

- The first line in the file corresponds to checking the “If you are specifying the ALOM password below” box in the plan variables section of the browser interface.
- The second line in the file is the password to access ALOM.
- The third line in the file corresponds to checking the “If you are specifying the terminal password below” box in the plan variables section of the browser interface.
- The fourth line in the file is the password to access the terminal.

Make sure to name the file something useful so that you can remember it for the next step. For example, you might name the file `/tmp/testv240-target`.

3. **To run the plan and create the target, type a command similar to the following example:**

```
# cat /tmp/ALOM-target | cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/targets/SunALOM-create -tar H:NM:masterserver-osp \
-comp + -vs testv240 -pto 30 -nto 10 -f /tmp/testv240-target
```

Defining a Sun IPMI Target Host

The Sun IPMI target host type is used for systems such as a Sun Fire™ V20z or V60z system.

▼ How to Create a Sun IPMI Target Host (Browser Interface)

For this task, the target host is a Sun Intelligent Platform Management Interface (IPMI) system. Specifically, the example defines a V20z target host.

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the Sun IPMI Targets sub-section of the Target Hosts section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the SunIPMI component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the SunIPMI component row of the Plan Parameters table.**

a. **Click Create Set.**

b. **Type a name for the variables set.**

c. **Verify variables and change them, if needed.**

   The prompts in the component variables table are mostly self-explanatory. However, if you need more information, see the detailed list of variables, along with default values and examples in "Sun IPMI Target Host Variables" on page 115.

d. **Decide whether to enter passwords in the variables set or in the plan prompts.**

   There are two methods for entering the passwords:

   - Type appropriate values in the following variables:

     ```
     ipmi_access_userid
     ipmi_access_password
     ssh_access_userid
     ssh_access_password
     ```

     For information about encrypting the passwords, see "Password Encryption" on page 113.
   - Leave the component variables blank and enter the password in the plan prompts provided in the plan run page.

e. **Save the variables set.**

f. **Select the variables set that you just saved from the drop-down menu in the SunIPMI component row of the Plan Parameters table.**

5. **From the Plan Details Run page, select the provisioning server on which to define the target host.**

Tip – The host name of the OS provisioning server ends in `-osp`.

---

6. **To encrypt the password from the plan, follow these steps on the Plan Details Run page.**

a. **Click the checkbox next to "If you are specifying the IPMI password below."**

b. **Type the password in the Password to Access IPMI field.**

   The password encrypts as you type.

c. **Click the checkbox next to "If you are specifying the SSH password below."**

d. **Type the password in the SSH Password to Access the Remote Management Interface field.**

   The password encrypts as you type.

7. **Click Run Plan (includes preflight).**

8. **To verify that the target host is created successfully, click the Hosts link in the left side of the N1 Grid SPS browser interface.**

   You should see the name of the virtual host appended with `-target`. For example, if you defined a provisioning target for host `sunfire-v60x`, the virtual host is `sunfire-v20z-target`.

▼ How to Create a Sun IPMI Target Host (Command-Line Interface)

Steps

1. **To create a variable set for the SunIPMI component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/targets/SunIPMI -name "v20z" \
-u admin -p admin -vars "installPath=v20zhost;description=test v20z host; \
ethernet_mac_address=00:09:3d:10:94:a6;ethernet_ip_address=10.42.42.110; \
ethernet_netmask=255.255.255.0;architecture=x86;kernel_arch=i86pc; \
guid={3AE45192 159D 11D8 AEF3 001083FDFE90};disk_size=20GB;ipmi_ip_address=10.5.133.134; \
ipmi_access_userid;ipmi_access_password=;ssh_access_userid=admin;ssh_access_password="
```

   For more information about the variables, see "Sun IPMI Target Host Variables" on page 115.

2. **To set up the appropriate password access, create a file that contains the following entries:**

```
true
admin
true
admin
```

   Where:

   - The first line in the file corresponds to checking the "If you are specifying the IPMI password below" box in plan variables section of the browser interface.
   - The second line in the file is the password to access IPMI.
   - The third line in the file corresponds to checking the "If you are specifying the SSH password below" box in plan variables section of the browser interface.
   - The fourth line in the file is the SSH password to access the remote management interface.

   Make sure to name the file something useful so that you can remember it for the next step. For example, you might name the file `/tmp/testv20z-target`.

3. **To run the plan and create the V20z target, type a command similar to the following example:**

```
# cat /tmp/testv20z-target | cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/targets/SunIPMI-create -tar H:NM:masterserver-osp \
-comp + -vs v20z -pto 30 -nto 10 -f /tmp/test-v20z
```

Defining an HP ILO Target Host

The following tasks show how to define an HP ILO host, for example an HP ProLiant series system.

▼ How to Create an HP ILO Target Host (Browser Interface)

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the HP ILO Targets sub-section of the Target Hosts section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the HPILO component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the HPILO component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Verify variables and change them, if needed.**

     The prompts in the component variables table are mostly self-explanatory. However, if you need more information, see the detailed list of variables, along with default values and examples in "HP ILO Target Host Variables" on page 121.

     d. **Decide whether to provide passwords in the variables set or in the plan prompts.**

     There are two methods for providing the passwords:

     - Type appropriate encrypted values in the following variables:

       ```
       ilo_access_userid
       ```

`ilo_access_password`

For information about encrypting the passwords, see "Password Encryption" on page 113.

- Leave the component variables blank and enter the password in the plan prompts provided in the plan run page.

e. **Save the variables set.**

f. **Select the variables set that you just saved from the drop-down menu in the HPILO component row of the Plan Parameters table.**

5. **On the Plan Details Run page, select the provisioning server on which to define the target host.**

Tip – The host name of the OS provisioning server ends in `-osp`.

---

6. **To encrypt the password from the plan, follow these steps on the Plan Details Run page.**

   a. **Click the checkbox next to "If you are specifying the ILO password below."**

   b. **Type the password in the Password to Access ILO field.**

   The password encrypts as you type.

7. **Click Run Plan (includes preflight).**

8. **To verify that the target host is created successfully, click the Hosts link in the left side of the N1 Grid SPS browser interface.**

   You should see the name of the virtual host appended with `-target`. For example, if you defined a provisioning target for host `hp-proliant-1`, the virtual host is `hp-proliant-1-target`.

▼ How to Create an HP ILO Target Host (Command-Line Interface)

Steps 1. **To create a variable set for the HPILO component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/targets/HPILO -name "hpProliant" \
-u admin -p admin -vars "installPath=HP-Proliant;description=Test description; \
ethernet_mac_address=00:0E:7F:FD:85:26;ethernet_ip_address=10.42.42.20; \
ethernet_netmask=255.255.255.0;architecture=x86;kernel_arch=i86pc; \
guid={4D303435-4C47-5037-3239-202020202020};disk_size=20GB;ilo_ip_address=10.5.133.166; \
ilo_version=1.6.4;ilo_access_userid=admin;ilo_access_password="
```

For more information about the variables, see “HP ILO Target Host Variables” on page 121.

2. **To set up the appropriate password access, create a file that contains the following entries:**

   ```
   true
   admin123
   ```

   Where:

   - The first line in the file corresponds to checking the “If you are specifying the ILO password below” box in plan variables section of the browser interface.
   - The second line in the file is the password to access ILO.

   Make sure to name the file something useful so that you can remember it for the next step. For example, you might name the file /tmp/hp-proliant.

3. **To run the plan and create the target host, type a command similar to the following example:**

```
# cat /tmp/hp-proliant | cr_cli -cmd pe.p.run -u admin -p admin  \
-PID NM:/com/sun/n1osp/targets/HPILO-create -tar H:NM:masterserver-osp -comp + \
-vs hpProliant -pto 30 -nto 10 -f /tmp/hp-proliant
```

Defining an Other Remotely-Managed Target Host

Use the remotely-managed target host type to identify target hosts that do not match any of the predefined types but that have other forms of remote management, such as SunFire B100s and B100x blades. The option provides the greatest flexibility of providing information regarding a target but also requires several variables to be filled.

There are two sets of remote management information gathered for this type of target. The first set of remote management information ends with _1 suffix, and the second set ends with the _2 suffix. Each remote network management information set requires information about the type of network management and the associated JAR file that handles the remote network management. The JAR files are located at /opt/SUNWn1osp/drivers on the OS provisioning server. Each remote network management interface can have a terminal server connection, in which case the information about the terminal server also has to be entered. Depending on the type of network management, the network management might support power management, boot mode, and console support. See the example below for more information.

▼ How to Create an Other Remotely-Managed Target Host (Browser Interface)

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the Other Remotely-Managed Targets sub-section of the Target Hosts section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the Other component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the Other component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Verify variables and change them, if needed.**

        The prompts in the component variables table are mostly self-explanatory. However, if you need more information, see the detailed list of variables, along with default values and examples in "Other Remotely-Managed Target Host Variables" on page 123.

     d. **Decide whether to provide passwords in the variables set or in the plan prompts.**

        There are two methods for providing the passwords:

        - Type appropriate encrypted values in the following variables:

          ```
          netmgmt_password_1
          terminal_server_access_password_1
          netmgmt_password_2
          terminal_server_access_password_2
          ```

          For information about encrypting the passwords, see "Password Encryption" on page 113.

        - Leave the component variables blank and enter the password in the plan prompts provided in the plan run page.

     e. **Save the variables set.**

     f. **Select the variables set that you just saved from the drop-down menu in the Other component row of the Plan Parameters table.**

5. **From the Plan Details Run page, select the provisioning server on which to define the target host.**

Tip – The host name of the OS provisioning server ends in `-osp`.

6. **To encrypt the password from the plan, follow these steps on the Plan Details Run page.**

   a. **Click the checkbox next to the first "If you are specifying the password below."**

   b. **Type the password to access the first network management interface in the Password to Access Management Interface 1 field.**

      The password encrypts as you type.

   c. **Click the checkbox next to the first "If you are specifying the terminal server password below."**

   d. **Type the password to access the first terminal server in the Password to Access Terminal Server Password 1 field.**

      The password encrypts as you type.

   e. **Click the checkbox next to the second "If you are specifying the password below."**

   f. **Type the password to access the second network management interface in the Password to Access Management Interface 2 field.**

      The password encrypts as you type.

   g. **Click the checkbox next to the second "If you are specifying the terminal server password below."**

   h. **Type the password to access the second terminal server in the Password to Access Terminal Server Password 2 field.**

      The password encrypts as you type.

7. **Click Run Plan (includes preflight).**

8. **To verify that the target host is created successfully, click the Hosts link in the left side of the N1 Grid SPS browser interface.**

   You should see the name of the virtual host appended with `-target`. For example, if you defined a provisioning target for host `sunfire-v60x`, the virtual host is `sunfire-v60x-target`.

▼ How to Create an Other Remotely-Managed Target Host (Command-Line Interface)

The following CLI example creates a target host for a SunFire B100s system.

Steps 1. **To create a variable set for the Other component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/targets/Other -name "b100s10" \
-u admin -p admin -vars "installPath=Blade-S10;description=S10 b100s blade target; \
ethernet_mac_address=00:03:ba:29:e4:ea;ethernet_ip_address=10.216.0.11; \
ethernet_netmask=255.255.255.192;architecture=sun4u;kernel_arch=sun4u;disk_size=30GB; \
netmgmt_type_1=b1600sc;netmgmt_version_1=1.0; \
netmgmt_hal_jar_1="file:///opt/SUNWn1osp/drivers/sun/b1600sc.jar"; \
netmgmt_ip_address_1=10.216.0.60;netmgmt_protocol_1=telnet;netmgmt_port_1=s10; \
power_service_true_false_1=true;boot_service_true_false_1=true; \
console_service_true_false_1=true;netmgmt_userid_1=admin;netmgmt_password_1=Clz6pK2b6qw="
```

For more information about the variables, see "Other Remotely-Managed Target Host Variables" on page 123.

2. **To set up the appropriate password access, create a file that contains the following entries:**

```
false
admin
false
admin
false
admin
false
admin
```

Where:

- The first line in the file corresponds to the first "If you are specifying the password below" box in the plan variables section of the browser interface.
- The second line in the file is the password to the first management interface.
- The third line in the file corresponds to the first "If you are specifying the terminal server password below" box in the plan variables section of the browser interface.
- The fourth line in the file is the password to the first terminal server..
- The fifth line in the file corresponds to the second "If you are specifying the password below" box in the plan variables section of the browser interface.
- The sixth line in the file is the password to the second management interface.
- The seventh line in the file corresponds to the second "If you are specifying the terminal server password below" box in the plan variables section of the browser interface.
- The eighth line in the file is the password to the second terminal server..

Make sure to name the file something useful so that you can remember it for the next step. For example, you might name the file `/tmp/b100s`.

3. **To run the plan and create the target host, type a command similar to the following example:**

```
# cat /tmp/b100s | cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/targets/Other-create -tar H:NM:masterserver-osp -comp + \
-vs b100s10 -pto 30 -nto 10 -f /tmp/b100s
```

Defining a Generic Target Host

You can manually power off and power on targets that do not have any remote management interfaces, such as workstations or desktop systems. To identify these targets for OS provisioning, use the Generic Target Host option.

▼ How to Create a Generic Target Host (Browser Interface)

Steps

1. **In the Common Tasks section of the N1 Grid SPS browser interface, select OS Provisioning.**

2. **On the OS Provisioning Common Tasks page, click Create in the Generic Targets sub-section of the Target Hosts section.**

3. **On the Plans Details page, click Run.**

4. **Choose variables to use for this plan.**

   - **To use an existing variables set, select a name from the drop-down menu in the Target component row of the Plan Parameters table.**

   - **To create a new variables set, click Select from List in the Target component row of the Plan Parameters table.**

     a. **Click Create Set.**

     b. **Type a name for the variables set.**

     c. **Determine whether to call the OSP Control Service.**

        For a generic target host, pay special attention to the `osp_control_service` variable. If you set this variable to `TRUE` and the system has a manual HAL, the N1 Grid SPS software stops and waits for you to perform the appropriate function. If you monitor the status of the installation through the status feature, the OS provisioning control service lets you know when to power off, reboot, and power on the target host.

        If the target host is on a private network and has no access to DHCP services, set this variable to `FALSE`.

     d. **Verify other variables and change them, if needed.**

        The other prompts in the component variables table are mostly self-explanatory. However, if you need more information, see the detailed list of variables, along with default values and examples in "Generic Target Host Variables" on page 127.

e. **Save the variables set.**

f. **Select the variables set that you just saved from the drop-down menu in the Target component row of the Plan Parameters table.**

5. **On the Plan Details Run page, select the provisioning server on which to define the target host.**

Tip – The host name of the OS provisioning server ends in `-osp`.

6. **Click Run Plan (includes preflight).**

7. **To verify that the target host is created successfully, click the Hosts link in the left side of the N1 Grid SPS browser interface.**

   You should see the name of the virtual host appended with `-target`. For example, if you defined a provisioning target for host `sunfire-v60x`, the virtual host is `sunfire-v60x-target`.

▼ How to Create a Generic Target Host (Command-Line Interface)

Steps

1. **To create a variable set for the Target component, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/targets/Target -name "mysystem" \
-u admin -p admin -vars "installPath=Generic;target_host=:[installPath]-target; \
description=Test description;ethernet_mac_address=00:0E:7F:FD:85:26; \
ethernet_ip_address=10.42.42.20;ethernet_netmask=255.255.255.0;architecture=x86; \
kernel_arch=i86pc;disk_size=20GB"
```

   For more information about the variables, see "Generic Target Host Variables" on page 127.

2. **To run the plan and create the target host, type a command similar to the following example:**

```
# cr_cli -cmd pe.p.run -u admin -p admin  -PID \
NM:/com/sun/n1osp/targets/Target-create -tar H:NM:masterserver-osp -comp + \
-vs mysystem -pto 30 -nto 10
```

Password Encryption

To provide a password for root access on a Solaris or Linux system, you need to use the operating system tools, rather than the OS provisioning tools, to encrypt the password.

Encrypting Passwords for OS Provisioning Variables

There are 3 ways to encrypt the password for OS provisioning variables:

- On the OS provisioning server, run the following command: `/opt/n1osp/sbin/n1osp_encrypter` *clear-text*. The encrypted version of the *clear-text* displays on the screen.
- Using the N1 Grid SPS browser interface, navigate to the plan `/com/sun/n1osp/untyped/EncryptPassword`. In that plan, type the password. To see the encrypted value, navigate through the progress details links and view the `stdout`.
- Type the following commands on the N1 Grid SPS server:

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/n1osp/untyped/EncryptPassword \
-tar H:NM:masterserver-osp -pto 30 -nto 10
# cr_cli -cmd pe.p.lo -u admin -p admin -ID id-from-the-EncryptPassword-command
# cr_cli -cmd pe.p.en -u admin -p admin -ID last-id-from-the-pe.p.lo-command
```

Encrypting Passwords for OS Root Access

If you need to provide a password for root access on a Solaris or Linux system, you need to use the operating system tools, rather than the OS provisioning tools, to encrypt the password.

Tools differ by operating system and by specific version of the operating system. On a Solaris 9 system, for example, most users would simply copy the root password from the `/etc/shadow` file. Consult your operating system documentation for more information.

APPENDIX **A**

Target Host Variables

Each type of target host has a number of specific configuration variables. This appendix contains a series of tables that explain the variables that apply to each target host type.

Sun IPMI Target Host Variables

The following table lists all component variables for the Sun IPMI target host. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE A–1 Component Variables for Sun IPMI Target Host

Variable Name	Description	Default or Sample Value
`installPath`	Physical host name for the target system	`sunfire-v60x-host` (Example)
`target_host`	Virtual host name for the target system	`:[installPath]-target` (Default)
`description`	(Optional) Description of the target	`test v60x` (Example)
`ethernet_mac_address`	Mac address	`00:0E:0C:07:E9:31` (Example)
`ethernet_ip_address`	IP address	`10.42.42.77` (Example)
`ethernet_netmask`	Network mask	`255.255.255.0` (Default)
`architecture`	Target architecture (for example, sun4u, x86)	`x86` (Default)
`kernel_arch`	Kernel architecture (sun4u, i86pc)	`i86pc` (Default)
`disk_size`	Size of disk	`20GB` (Example)

TABLE A–1 Component Variables for Sun IPMI Target Host *(Continued)*

Variable Name	Description	Default or Sample Value
`osp_control_service`	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)	`TRUE` (Default)
`sps_ra_parent`	Name of parent for Remote Agent (RA)	`masterserver` (Default)
`sps_ra_parent_type`	Whether RA parent is an MS (masterserver) or an LD (local distributor)	`MS` (Default)
`sps_ra_host`	Name or IP address for RA host	`:[installPath]` (Default)
`sps_ra_conn`	Type of RA connection (raw, ssl, ssh)	`raw` (Default)
`sps_ra_ssl_auth`	Whether RA has SSL authorization (y,n)	`N` (Default)
`sps_ra_port`	Port for RA	`1131` (Default)
`sps_ra_adv_parms`	Advanced parameters for RA	
`guid`	GUID of the target specified within {}	
`ipmi_ip_address`	IPMI IP address	`10.5.133.135` (Example)
`ipmi_access_userid`	User ID to access IPMI	
`ipmi_access_password`	Encrypted Password to access IPMI	
`ssh_access_userid`	ssh user ID to access the remote management interface	
`ssh_access_password`	Encrypted ssh password to access the remote management interface	

Sun ALOM Target Host Variables

The following table lists all component variables for the Sun ALOM target host. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE A–2 Component Variables for Sun ALOM Target Host

Variable Name	Description	Default or Sample Value
`installPath`	Physical host name for the target system	`sunfire-v60x-host` (Example)

TABLE A–2 Component Variables for Sun ALOM Target Host *(Continued)*

Variable Name	Description	Default or Sample Value
`target_host`	Virtual host name for the target system	`:[installPath]-target` (Default)
`description`	(Optional) Description of the target	`test v60x` (Example)
`ethernet_mac_address`	Mac address	`00:0E:0C:07:E9:31` (Example)
`ethernet_ip_address`	IP address	`10.42.42.77` (Example)
`ethernet_netmask`	Network mask	`255.255.255.0` (Default)
`architecture`	Target architecture (for example, sun4u, x86)	`sun4u` (Default)
`kernel_arch`	Kernel architecture (sun4u, i86pc)	`sun4u` (Default)
`disk_size`	Size of disk	`20GB` (Default)
`osp_control_service`	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)	`TRUE` (Default)
`sps_ra_parent`	Name of parent for RA	`masterserver` (Default)
`sps_ra_parent_type`	Whether RA parent is an MS or an LD	`MS` (Default)
`sps_ra_host`	Name or IP address of RA host	`:[installPath]` (Default)
`sps_ra_conn`	Type of RA connection (raw, ssl, ssh)	`raw` (Default)
`sps_ra_ssl_auth`	Whether RA has SSL authorization (y,n)	`N` (Default)
`sps_ra_port`	Port for RA	`1131` (Default)
`sps_ra_adv_parms`	Advanced parameters for RA	
`alom_ip_address`	ALOM IP address	`10.5.133.135` (Example)
`alom_access_userid`	User id to access ALOM	
`alom_access_password`	Encrypted password to access ALOM	
`terminal_server_ip_address`	IP address of terminal server connected to this system	`10.5.133.135` (Example)
`terminal_server_port`	Port of terminal server	
`terminal_server_access_userid`	Terminal server user id	
`terminal_server_access_password`	Encrypted password for terminal server	
`terminal_server_type`	Terminal server type	`Cisco2621` (Default)

TABLE A–2 Component Variables for Sun ALOM Target Host *(Continued)*

Variable Name	Description	Default or Sample Value
terminal_server_version	Terminal server version	
terminal_server_hal_jar	Path to terminal server hardware abstraction layer (HAL) JAR file	file:///opt/n1osp/drivers/cisco/c2621.jar (Default)

Sun LOM Target Host Variables

The following table lists all component variables for the Sun LOM target host. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE A–3 Component Variables for Sun LOM Target Host

Variable Name	Description	Default or Sample Value
installPath	Physical host name for the target system	sunfire-v60x-host (Example)
target_host	Virtual host name for the target system	:[installPath]-target (Default)
description	(Optional) Description of the target	test v60x (Example)
ethernet_mac_address	Mac address	00:0E:0C:07:E9:31 (Example)
ethernet_ip_address	IP address	10.42.42.77 (Example)
ethernet_netmask	Network mask	255.255.255.0 (Default)
architecture	Target architecture (for example, sun4u, x86)	sun4u (Default)
kernel_arch	Kernel architecture (sun4u, i86pc)	sun4u (Default)
disk_size	Size of disk	20GB (Default)
osp_control_service	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)	TRUE (Default)
sps_ra_parent	Name of parent for RA	masterserver (Default)
sps_ra_parent_type	Whether RA parent is an MS or an LD	MS (Default)
sps_ra_host	Name or IP address of RA host	
sps_ra_conn	Type of RA connection (raw, ssl, ssh)	raw (Default)
sps_ra_ssl_auth	Whether RA has SSL authorization (y,n)	N (Default)

TABLE A–3 Component Variables for Sun LOM Target Host *(Continued)*

Variable Name	Description	Default or Sample Value
`sps_ra_port`	Port for RA	`1131` (Default)
`sps_ra_adv_parms`	Advanced parameters for RA	
`disk_type`	SCSI or IDE	`scsi` (Default)
`lom_version`	LOM version	
`lom_hal_jar`	Path to LOM HAL JAR file	`file:///opt/n1osp/drivers/sun/lom.jar` (Default)
`terminal_server_ip_address`	IP address of terminal server connected to this system	`10.5.133.135` (Example)
`terminal_server_port`	Port of terminal server	
`terminal_server_userid`	Terminal server user id	
`terminal_server_password`	Encrypted value for terminal server password	
`terminal_server_type`	Terminal server type	`Cisco2621` (Default)
`terminal_server_version`	Terminal server version	
`terminal_server_hal_jar`	Path to terminal server HAL JAR file	`file:///opt/n1osp/drivers/cisco/c2621.jar` (Default)
`power_service_true_false`	Does the network management interface provide power on/off service	`true` (Default)
`boot_service_true_false`	Does the network management interface provide boot service (boot order disk/network)	`true` (Default)
`console_service_true_false`	Does the network management interface provide console	`false` (Default)

Sun Mid-Range Target Host Variables

The following table lists all component variables for the Sun mid-range target host. The `SunMidRange` component uses these variables. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE A–4 Component Variables for Sun Mid-Range Target Host

Variable Name	Description	Default or Sample Value
installPath	Physical host name for the target system	sunfire-v60x-host (Example)
target_host	Virtual host name for the target system	:[installPath]-target (Default)
description	(Optional) Description of the target	test v60x (Example)
ethernet_mac_address	Mac address	00:0E:0C:07:E9:31 (Example)
ethernet_ip_address	IP address	10.42.42.77 (Example)
ethernet_netmask	Network mask	255.255.255.0 (Default)
architecture	Target architecture (for example, sun4u, x86)	sun4u (Default)
kernel_arch	Kernel architecture (sun4u, i86pc)	sun4u (Default)
disk_size	Size of disk	20GB (Default)
osp_control_service	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)	TRUE (Default)
sps_ra_parent	Name of parent for RA	masterserver (Default)
sps_ra_parent_type	Whether RA parent is an MS or an LD	MS (Default)
sps_ra_host	Name or IP address for RA host	
sps_ra_conn	Type of RA connection (raw, ssl, ssh)	raw (Default)
sps_ra_ssl_auth	Whether RA has SSL authorization (y,n)	N (Default)
sps_ra_port	Port for RA	1131 (Default)
sps_ra_adv_parms	Advanced parameters for RA	
domain_name	Domain	
domainsc_access_userid	User id to access domain SC	
domainsc_access_password	Encrypted password to access domain SC	
platformsc_ip_address	IP address of platform SC	
platformsc_access_userid	User id to access platform SC	
platformsc_access_password	Encrypted password to access platform SC	

HP ILO Target Host Variables

The following table lists all component variables for the HP ILO target host. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE A–5 Component Variables for HP ILO Target Host

Variable Name	Description	Default or Sample Value
installPath	Physical host name for the target system	sunfire-v60x-host (Example)
target_host	Virtual host name for the target system	:[installPath]-target (Default)
description	(Optional) Description of the target	test v60x (Example)
ethernet_mac_address	Mac address	00:0E:0C:07:E9:31 (Example)
ethernet_ip_address	IP address	10.42.42.77 (Example)
ethernet_netmask	Network mask	255.255.255.0 (Default)
architecture	Target architecture (for example, sun4u, x86)	x86 (Default)
kernel_arch	Kernel architecture (sun4u, i86pc)	i86pc (Default)
disk_size	Size of disk	20GB (Default)
osp_control_service	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)	TRUE (Default)
sps_ra_parent	Name of parent for RA	masterserver (Default)
sps_ra_parent_type	Whether RA parent is an MS or an LD	MS (Default)
sps_ra_host	Name or IP address of RA host	:[installPath] (Default)
sps_ra_conn	Type of RA connection (raw, ssl, ssh)	raw (Default)
sps_ra_ssl_auth	Whether RA has SSL authorization (y,n)	N (Default)
sps_ra_port	Port for RA	1131 (Default)
sps_ra_adv_parms	Advanced parameters for RA	
guid	GUID of the target specified within {}	{3AE45192 159D 11D8 AEF3 001083FDFE8A} (Example)
ilo_ip_address	ILO IP address	10.5.133.135 (Example)
ilo_version	ILO version	1.6.4 (Default)

TABLE A–5 Component Variables for HP ILO Target Host *(Continued)*

Variable Name	Description	Default or Sample Value
`ilo_access_userid`	User id to access ILO	
`ilo_access_password`	Encrypted password to access ILO	

Other IPMI Target Host Variables

The following table lists all component variables for the other IPMI target host. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE A–6 Component Variables for Other IPMI Target Host

Variable Name	Description	Default or Sample Value
`installPath`	Physical host name for the target system	`sunfire-v60x-host` (Example)
`target_host`	Virtual host name for the target system	`:[installPath]-target` (Default)
`description`	(Optional) Description of the target	`test v60x` (Example)
`ethernet_mac_address`	Mac address	`00:0E:0C:07:E9:31` (Example)
`ethernet_ip_address`	IP address	`10.42.42.77` (Example)
`ethernet_netmask`	Network mask	`255.255.255.0` (Default)
`architecture`	Target architecture (for example, sun4u, x86)	`x86` (Default)
`kernel_arch`	Kernel architecture (sun4u, i86pc)	`i86pc` (Default)
`disk_size`	Size of disk	`20GB` (Default)
`osp_control_service`	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)	`TRUE` (Default)
`sps_ra_parent`	Name of parent for RA	`masterserver` (Default)
`sps_ra_parent_type`	Whether RA parent is an MS or an LD	`MS` (Default)
`sps_ra_host`	Name or IP address of RA host	`:[installPath]` (Default)
`sps_ra_conn`	Type of RA connection (raw, ssl, ssh)	`raw` (Default)

TABLE A–6 Component Variables for Other IPMI Target Host *(Continued)*

Variable Name	Description	Default or Sample Value
`sps_ra_ssl_auth`	Whether RA has SSL authorization (y,n)	`N` (Default)
`sps_ra_port`	Port for RA	`1131` (Default)
`sps_ra_adv_parms`	Advanced parameters for RA	
`guid`	GUID of the target specified within {}	{`3AE45192 159D 11D8 AEF3 001083FDFE8A`} (Example)
`disk_type`	Disk type (for example, SCSI, IDE)	`scsi` (Default)
`ipmi_ip_address`	IPMI IP address	`10.5.133.135` (Example)
`ipmi_access_userid`	User id to access IPMI	
`ipmi_access_password`	Encrypted password to access IPMI	
`ipmi_hal_jar`	Path to IPMI HAL JAR file	`file:///opt/n1osp/drivers/sun/ipmi.jar` (Default)
`ipmi_version`	IPMI version	`1.5` (Default)
`power_service_true_false`	Does the network management interface provide power on/off service?	`true` (Default)
`boot_service_true_false`	Does the network management interface provide boot service (disk, then network)?	`false` (Default)
`console_service_true_false`	Does the network management interface provide console service?	`false` (Default)

Other Remotely-Managed Target Host Variables

The following table lists all component variables for the other remotely-managed target host. The `Other` component type uses these variables. Use the `Other` type for remotely-managed targets that do not fit any of the other categories. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

Note – The `Other` component type has a long list of variables, some of which will not apply to your specific hardware type. If a variable does not apply, leave it blank.

TABLE A–7 Component Variables for Other Remotely-Managed Target Hosts

Variable Name	Description	Default Value (if supplied)
`installPath`	Physical host name for the target system	`sunfire-v60x-host` (Example)
`target_host`	Virtual host name for the target system	`:[installPath]-target` (Default)
`description`	(Optional) Description of the target	`test v60x` (Example)
`ethernet_mac_address`	Mac address	`00:0E:0C:07:E9:31` (Example)
`ethernet_ip_address`	IP address	`10.42.42.77` (Example)
`ethernet_netmask`	Network mask	`255.255.255.0` (Default)
`architecture`	Target architecture (for example, sun4u, x86)	`x86` (Default)
`kernel_arch`	Kernel architecture (sun4u, i86pc)	`i86pc` (Default)
`disk_size`	Size of disk	`20GB` (Default)
`osp_control_service`	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)	`TRUE` (Default)
`sps_ra_parent`	Name of parent for RA	`masterserver` (Default)
`sps_ra_parent_type`	Whether RA parent is an MS or an LD	`MS` (Default)
`sps_ra_host`	Name or IP address of RA host	`:[installPath]` (Default)
`sps_ra_conn`	Type of RA connection (raw, ssl, ssh)	`raw` (Default)
`sps_ra_ssl_auth`	Whether RA has SSL authorization (y,n)	`N` (Default)
`sps_ra_port`	Port for RA	`1131` (Default)
`sps_ra_adv_parms`	Advanced parameters for RA	
`guid`	GUID of the target specified within {}	{`3AE45192 159D 11D8 AEF3 001083FDFE8A`} (Example)
`disk_type`	Disk type (for example, SCSI, IDE)	`scsi` (Default)
`netmgmt_type_1`	Network management interface type (ipmi/alom/lom)	
`netmgmt_version_1`	Network management version	

TABLE A–7 Component Variables for Other Remotely-Managed Target Hosts *(Continued)*

Variable Name	Description	Default Value (if supplied)
`netmgmt_hal_jar_1`	Path to network management HAL JAR file	`file:///opt/n1osp/drivers/sun/ipmi.jar` (Default)
`netmgmt_ip_address_1`	Network management IP address	
`netmgmt_protocol_1`	Network management protocol (ipmi/telnet)	`telnet` (Default)
`netmgmt_port_1`	Network management port	
`power_service_true_false_1`	Does the network management interface provide power on/off service?	`true` (Default)
`boot_service_true_false_1`	Does the network management interface provide boot service (disk, then network)?	`false` (Default)
`console_service_true_false_1`	Does the network management interface provide console service?	`false` (Default)
`netmgmt_userid_1`	User ID to access network management interface	
`netmgmt_password_1`	Encrypted password to access network management interface	
`terminal_server_type_1`	Terminal server type	`Cisco2621` (Default)
`terminal_server_version_1`	Terminal server version	
`terminal_server_hal_jar_1`	Path to terminal server HAL JAR file	`file:///opt/n1osp/drivers/cisco/c2621.jar` (Default)
`terminal_server_ip_address_1`	IP address of terminal server connected to this system	`10.5.133.135` (Example)
`terminal_server_protocol_1`	Terminal server access protocol	
`terminal_server_port_1`	Port of terminal server	
`terminal_server_access_userid_1`	Terminal server user id	
`terminal_server_access_password_1`	Encrypted password for terminal server	
`netmgmt_type_2`	Network management interface type (ipmi/alom/lom)	
`netmgmt_version_2`	Network management version	

TABLE A–7 Component Variables for Other Remotely-Managed Target Hosts *(Continued)*

Variable Name	Description	Default Value (if supplied)
`netmgmt_hal_jar_2`	Path to network management HAL JAR file	`file:///opt/n1osp/drivers/sun/ipmi.jar` (Default)
`netmgmt_ip_address_2`	Network management IP address	
`netmgmt_protocol_2`	Network management protocol (ipmi/telnet)	`telnet` (Default)
`netmgmt_port_2`	Network management port	
`power_service_true_false_2`	Does the network management interface provide power on/off service?	`true` (Default)
`boot_service_true_false_2`	Does the network management interface provide boot service (disk, then network)?	`false` (Default)
`console_service_true_false_2`	Does the network management interface provide console service?	`false` (Default)
`netmgmt_userid_2`	User ID to access network management interface	
`netmgmt_password_2`	Encrypted password to access network management interface	
`terminal_server_type_2`	Terminal server type	`Cisco2621` (Default)
`terminal_server_version_2`	Terminal server version	
`terminal_server_hal_jar_2`	Path to terminal server HAL JAR file	`file:///opt/n1osp/drivers/cisco/c2621.jar` (Default)
`terminal_server_ip_address_2`	IP address of terminal server connected to this system	`10.5.133.135` (Example)
`terminal_server_protocol_2`	Terminal server access protocol	
`terminal_server_port_2`	Port of terminal server	
`terminal_server_access_userid_2`	Terminal server user id	
`terminal_server_access_password_2`	Encrypted password for terminal server	

Generic Target Host Variables

The following table lists all component variables for the generic target host. The Target type uses these variables. Use the Target type for any host that does not match any of the other target host types. For those variables that have default values, the default is listed. For those variables that do not have default values, an example value is shown for illustration.

TABLE A–8 Component Variables for Generic Target Host

Variable Name	Description	Default or Sample Value
installPath	Physical host name for the target system	
target_host	Virtual host name for the target system	:[installPath]-target (Default)
description	(Optional) Description of the target	test v60x (Example)
ethernet_mac_address	Mac address	00:0E:0C:07:E9:31 (Example)
ethernet_ip_address	IP address	10.42.42.77 (Example)
ethernet_netmask	Network mask	255.255.255.0 (Default)
architecture	Target architecture (for example, sun4u, x86)	sun4u (Default)
kernel_arch	Kernel architecture (sun4u, i86pc)	sun4u (Default)
disk_size	Size of disk	20GB (Default)
osp_control_service	Whether to call the OSP Control Service, which automates power off, reboot, and power on functions.	TRUE (Default)
sps_ra_parent	Name of parent for RA	masterserver (Default)
sps_ra_parent_type	Whether RA parent is an MS or an LD	MS (Default)
sps_ra_host	Name or IP address of RA host	
sps_ra_conn	Type of RA connection (raw, ssl, ssh)	raw (Default)
sps_ra_ssl_auth	Whether RA has SSL authorization (y,n)	N (Default)
sps_ra_port	Port for RA	1131 (Default)
sps_ra_adv_parms	Advanced parameters for RA	

APPENDIX B

Solaris Profile Component Variables

The Solaris Profile component includes variables for each JET module. These variables are separated into the following categories:

- "Basic OS Configuration Variables" on page 129
- "Variables for N1 Grid SPS Remote Agents" on page 138
- "General OS Provisioning Plug-In Variables" on page 140

Basic OS Configuration Variables

base_config Variables

The variables described in the following table provide basic operating system configuration. These variables directly equate to the JET base_config module. In the table, you see several variables that use the value of the *targetable component*. For purposes of the Solaris Profile variables, the targetable component is the target host.

TABLE B–1 Component Variables for base_config Module

Variable name	Description	Default or Sample Value
ClientArch_base_config	Kernel architecture, such as sun4u or x86. By default, this is set to the kernel architecture of the targetable component.	:[targetableComponent:kernel_arch] (Default)

TABLE B–1 Component Variables for base_config Module *(Continued)*

Variable name	Description	Default or Sample Value
ClientEther_base_config	Ethernet MAC address. By default, this is set to the Ethernet MAC address of the targetable component.	:[targetableComponent: ethernet_mac_address] (Default)
ClientOS_base_config	Version of the OS to be provisioned	Solaris9_u7_sparc (Example)
client_allocation_base_config	The mechanism used to build this client. By default, the options listed in /opt/jet/etc/jumpstart.conf are used. Leave the value blank unless you need to do something different from the default for this specific client.	
products_base_config	JET modules to provision	base_config spsra (Example)
sysidcfg_nameservice_ base_config	Name service to configure at initial boot	NIS (Default)
sysidcfg_network_ interface_base_	Network interface to use at initial boot	PRIMARY (Default)
sysidcfg_ip_address_ base_config	IP address to use at initial boot. By default, this is set to the IP address of the targetable component	:[targetableComponent: ethernet_ip_address](Default)
sysidcfg_netmask_base_config	Netmask to use at initial boot. By default, this is set to the netmask of the targetable component	:[targetableComponent: ethernet_netmask](Default)
sysidcfg_root_password_base_config	Encrypted root password	boajrOmU7GFmY (Example)
sysidcfg_system_locale_ base_conf	System locale to use at initial boot.	en_US.ISO8859-1 (Example)
sysidcfg_timeserver_ base_config	Where to get system time for initial boot. If blank, system time comes from the JET server. Alternatively, you can set to localhost to get the system time from the hardware clock on the client	
sysidcfg_timezone_ base_config	System time zone to use for initial boot.	US/Pacific (Example)
sysidcfg_terminal_ base_config	Terminal emulator to set at initial boot.	vt100 (Default)
sysidcfg_security_policy_ base_co	Kerberos security policy to use at initial boot.	NONE (Default)

TABLE B–1 Component Variables for `base_config` Module *(Continued)*

Variable name	Description	Default or Sample Value
`sysidcfg_protocol_ipv6_base_conf`	Whether to use IPv6 protocol at initial boot.	`no` (Default)
`sysidcfg_default_route_base_conf`	Default route to set for Solaris 9 or later environments.	
`x86_nowin_base_config`	For x86 systems, prevents Solaris from trying to run Windows during the install.	`yes` (Default)
`x86_console_base_config`	For x86 systems, set the console to the correct tty port if you are not going to connect a keyboard and monitor to the client. Setting this variable enables you to perform installs via the serial port. For b1600, v20z, and v40z systems, use `ttya`. For lx50, v60x, and v65x systems, use `ttyb`.	
`x86_disable_acpi_base_config`	For x86 systems, any value disables ACPI. Disabling ACPI might make the installation process proceed better due to how the interrupts are handled.	
`x86_safetoreboot_base_config`	For x86 systems, controls whether the system automatically reboots. If your PXE boot is a one time option, and the next reboot will attempt to boot from disk, you should set this option to `yes`.	
`profile_base_config`	Create your own custom profile. Set this variable to reference a file relative to the `Clients/` directory. Otherwise, set the following 21 `profile_*` variables and the plug-in will create the profile.	

TABLE B–1 Component Variables for base_config Module *(Continued)*

Variable name	Description	Default or Sample Value
profile_cluster_ base_config	Solaris software group package. SUNWCreq identifies the basic, required software. Other possibilities include: ■ SUNWCuser — User package ■ SUNWCprog — User and developer packages ■ SUNWCall — All packages ■ SUNWCXall — All packages, including OEM packages (mandatory for E10K environments) ■ SUNWCrnet — Minimal package for Solaris 10 only	SUNWCreq (Default)
profile_usedisk_ base_config	Defines the boot disks onto which the OS will be loaded. Use the format c*n*t*n*d*n* or the keyword rootdisk. If the value is rootdisk. , then the current boot disk will be used.	rootdisk (Default)
profile_dontuse_ base_config	A comma-separated list of disks that should not be used. Use the format c*n*t*n*d*n*. This variable applies only if profile_usedisk_base_config is not set.	
profile_root_base_config	Root space (free, or MB size)	free (Default)
profile_swap_base_config	Swap space (MB size)	256 (Default)
profile_s3_mtpt_base_config	Mount path to the slice 3 partition. If you are using VxVM and you want your boot disk to look like the mirror, then leave slices 3 and 4 empty.	
profile_s3_size_base_config	Size of the slice 3 partition.	
profile_s4_mtpt_base_config	Mount path to the slice 4 partition.	
profile_s4_size_base_config	Size of the slice 4 partition.	
profile_s5_mtpt_base_config	Mount path to the slice 5 partition.	/var (Default)
profile_s5_size_base_config	Size of the slice 5 partition.	
profile_s6_mtpt_base_config	Mount path to the slice 6 partition.	/usr (Default)
profile_s6_size_base_config	Size of the slice 6 partition.	

TABLE B–1 Component Variables for `base_config` Module *(Continued)*

Variable name	Description	Default or Sample Value
`profile_s7_mtpt_base_config`	Mount path to the slice 7 partition. If you are using Solaris Volume Manager (SVM), the default behavior is to use slice 7 as a location for metastate databases. If you are using the SVM default configuration, do not use slice 7 for data.	`/opt` (Default)
`profile_s7_size_base_config`	Size of the slice 7 partition.	
`profile_additional_disks_base_co`	A comma-separated list of additional disks to use and configure. Use the format `c`*n*`t`*n*`d`*n*. For each disk listed, define sets of variables for each slice to identify the mount point and the size. Use a colon to separate the variables.	
`profile_add_locales_base_config`	Comma-separated list of locales to add.	`fr_FR, ja_JP.UTF-8` (Example)
`profile_del_locales_base_config`	Comma-separated list of locales to delete.	
`profile_add_geos_base_config`	Comma-separated list of geographical regions to add.	`N_Europe, C_Europe` (Example)
`profile_del_geos_base_config`	Comma-separated list of geographical regions to delete.	
`ufs_logging_filesys_base_config`	For Solaris 7 and later systems, a space-separated list of mount points to use for logging. To enable logging on all UFS file systems, use the keyword `all`. Solaris 9 09/04 enables logging by default. To disable logging on a specific file system, put a hyphen in front of the mount point. To disable logging on all file systems, use the keyword `none`. You cannot mix keywords and mount points. You can specify the root file system (/), although the root file system is included as part of the `all` and `none` keywords.	`all` (Default)
`profile_add_packages_base_config`	Space-separated list of packages to add.	

TABLE B–1 Component Variables for base_config Module *(Continued)*

Variable name	Description	Default or Sample Value
profile_del_packages_ base_config	Space-separated list of packages to remove.	
profile_add_clusters_ base_config	Space-separated list of cluster packages to add.	
profile_del_clusters_ base_config	Space-separated list of cluster packages to remove.	SUNWCpm SUNWCpmx SUNWCdial SUNWCdialx (Example)
nfs_mounts_base_config	Space-separated list of remote NFS mount points. Use ? to separate the mount source from the mount target, as shown in the example.	fs?1.1.1.1:/fs (Example)
nodename_base_config	Value to use for /etc/nodename if not the default hostname.	
defaultrouter_base_config	Value to use for /etc/defaultrouter.	
notrouter_base_config	If set to y, then disable IPv4 forwarding and create /etc/notrouter file.	
dns_domain_base_config	DNS domain entry for /etc/resolv.conf file.	
dns_nameservers_base_config	Space-separated list of IP addresses to use for DNS nameserver entries in /etc/resolv.conf file.	
dns_searchpath_base_config	List of entries to go in the DNS search line in /etc/resolv.conf file.	
dns_disableforbuild_ base_config	Delay DNS configuration until later. If DNS is not available in the build environment, set this variable to yes.	
ntp_servers_base_config	Space-separated list of names or IP addresses for the NTP servers. The first server will be given a 'prefer' tag. This section only place lines of the form: server [prefer] into the /etc/inet/ntp.conf file. For additional NTP control, use the custom module to deploy your own custom ntp.conf file.	

TABLE B–1 Component Variables for `base_config` Module *(Continued)*

Variable name	Description	Default or Sample Value
`networkifs_base_config`	Space-separated list of interfaces to be defined. For logical interfaces, use underscores (_) rather than colons (`:`). Use the format *cntndn*. For each interface listed, define sets of variables to provide the netname, netmask, hostname, and IP address for the interface. Use a colon to separate the variables.	`le0 hme0` (Example)
`ipmp_networkifs_base_config`	Space-separated list of interfaces to be defined under IPMP control. For each interface listed, define sets of variables to provide the netgroup, mode, test1, test2, netmask, hostname, log-ip, hostname2, and log-ip2 for the interface. Use a colon to separate the variables.	`qfe0_qfe4 qfe1_qfe5` (Example)
`update_terminal_base_config`	If set, put the `sysidcfg` terminal type into `inittab`.	`yes` (Default)
`enable_savecore_base_config`	If set to any value, enable save core for Solaris 2.6 systems.	`yes` (Default)
`dumpadm_minfree_base_config`	Set a limit so that crash dumps do not fill up the `dump` filesystem. See dumpadm(1M) -m option for possible values.	`20000k` (Example)
`noautoshutdown_base_config`	If set to any value, disable power management.	`pm_disabled` (Default)
`enable_rootlogin_base_config`	If set to any value, enable network root login from both telnet/rsh and ssh.	
`enable_rootftp_base_config`	If set to any value, enable root FTP access.	
`shutup_sendmail_base_config`	If set, create an alias hostname to shut up sendmail.	`yes` (Default)
`poweroff_afterbuild_base_config`	If set, shut down the machine once the build completes.	
`dedicated_dump_device_` `base_confi`	If set, the `dumpadm` utility configures the partition as a Dedicated Dump Device. See dumpadm(1M) for supported Operating Environments.	

TABLE B–1 Component Variables for base_config Module *(Continued)*

Variable name	Description	Default or Sample Value
enable_altbreak_base_config	If set, enable alternate break sequence.	
disable_sysid_probe_ base_config	If set, skip the sysid stuff on the first reboot. This can significantly increase provisioning efficiency on systems that have many unused network adapters.	yes (Default)
nfsv4_domain_base_config	Set up the NFSv4 domain to prevent being prompted at first reboot. If not set, look first for the entry in dns_domain_base_config, and second for the domain value in /etc/default/nfs .	
productdir_base_config	Path to the products. If blank, use information from the jumpstart.conf file and the IP address of the JET server. If your package files are not stored on the JET server, then provide an NFS-style path to the location of the packages.	
patchdir_base_config	Path to the patches. If blank, use information from the jumpstart.conf file and the IP address of the JET server. If your patch files are not stored on the JET server, then provide an NFS-style path to the location of the patches.	

Configuring Additional Slices

The profile_additional_disks_base_co variable provides a list of additional slices to configure and use. Use the format c*n*t*n*d*n* to identify each slice. For each slice, you provide a pair of variables that defines the mount point and size of the slice. Use a colon to separate the variables.

The following example shows the content of the profile_additional_disks_base_co variable to define two additional slices.

```
c0t0d0s3 /export/mydisk1:50MB
c0t1d1s3 /export/mydisk2:100MB
```

Configuring Network Interfaces

The networkifs_base_config variable provides a list of network interfaces to use. For logical interfaces, use underscores (_) rather than colons (:). For each interface listed, you provide a group of four variables that defines the netname, netmask, hostname, and IP address for the interface. Use a colon to separate the variables.

The following example shows the content of the networkifs_base_config variable to define two interfaces.

```
le0 bkp:255.255.255.0:me-bkp:192.168.1.0
hme0 bkp2:255.255.255.0:me-bkp2:192.168.2.0
```

Configuring IP Network Multipathing (IPMP)

For general information about IPMP, see the IP Network Multipathing Administration Guide.

The default mode for IP multipathing in Solaris 8 or later environments is automatic failback. To change this mode, edit the /etc/default/mpathd file.

You can only set up an IPMP group with pairs of interfaces in one of the following configurations:

- active-standby failover — Set ipmp mode = s, and specify one logical hostname/ip address pair.
- failover with outbound load spreading — Set ipmp mode = l, and specify one logical hostname/ip address pair.
- active-active with outbound load spreading — Set ipmp mode = l, specify a second logical hostname/ip address pair for the second interface.

The ipmp_networkifs_base_config variable defines a list of interfaces to be defined under IPMP control. For each interface, a set of variables must be included to defined additional interface parameters. The following lists the parameters that you need to define:

- netgroup — identifies the IPMP interface group name; for example, database-net
- mode — specifies the IPMP mode; Valid values include:
  - s — standby mode; used for failover only; In this mode, test addresses are allocated last. The first test address is on the first virtual interface of the first physical adapter. The second test address is on the second physical adapter.
  - l — load spreading; used for active/active mode. In this mode, test addresses are allocated on the first virtual interfaces on both the first and second physical adapters.

  To force the test addresses onto the physical adapters, add the suffix “p” to either mode parameter.

Caution – Be careful forcing the test addresses onto the physical adapter, as this might break some applications.

- test1 — IPMP test address 1

Note – This address must not be in the hosts file or otherwise used.

- test2 — IPMP test address 2

Note – This address must not be in the hosts file or otherwise used.

- netmask — Netmask for IPMP pair
- hostname — Unique hostname for logical IP
- log-ip — Logical IP for first i/f of pair
- hostname 2 — For active-active configurations only, unique hostname for logical IP
- log-ip2 — For active-active configurations only, local IP for second i/f of pair

The following example shows the content of the `ipmp_networkifs_base_config` variable to define one IPMP interface.

```
qfe0_qfe1 database-net:1:10.0.0.1:10.0.0.2:24:oracle-db:10.0.0.3:apache:10.0.0.4
```

Variables for N1 Grid SPS Remote Agents

The variables described in the following table provide Remote Agent (RA) configuration for the N1 Grid SPS software. These variables directly equate to the JET `spsra` module.

TABLE B–2 Component Variables for `spsra` Module

Variable name	Description	Example or Default
`install_spsra`	Whether to install the N1 Grid SPS RA software.	`y` (Default)

TABLE B–2 Component Variables for spsra Module *(Continued)*

Variable name	Description	Example or Default
product_version_spsra	Version of RA to install.	5.0 (Default)
basedir_spsra	Base directory in which RA files are installed.	/opt/SUNWn1sps (Default)
install_ra_from_snapshot_spsra	Determines where to get RA files. When you install the JET server or attach the JET server, a snapshot of the RA on the JET server is taken automatically. As a result, this variable defaults to y. Set the variable to n for a cross architecture install. In this case, you need to import the product media for the JET module by using the Manage link in the JET Solaris Image Servers section of the OS Provisioning common tasks page.	y (Default)
username_spsra	When using the RA installer, specify user name to own the RA filesystem.	jdoe (Example)
groupname_spsra	When using the RA installer, specify group name to own the RA filesystem.	staff (Example)
setup_user_acct_spsra	Determines whether to set up user and group account for username_spsra and groupname_spsra .	n (Default)
user_passwd_spsra	Encrypted password for username_spsra .	6GR9gOi1vRVFw (Example)
userid_spsra	User account ID for username_spsra.	505 (Example)
groupid_spsra	Group account ID for groupname_spsra.	505 (Example)
homedir_spsra	Home directory for RA files.	/opt/SUNWn1sps (Default)
jre_path_spsra	Path to the Java runtime environment. If no value is provided, use the path suggested by the installer. If a path is given, the JRE must be installed at that path or the installation fails.	
setuid_spsra	Enable RA root access.	y (Default)
conn_spsra	Determines method to use for connecting to RA. By default, uses the connection method defined for the targetable component.	:[targetableComponent:sps_ra_conn] (Default)

TABLE B–2 Component Variables for spsra Module *(Continued)*

Variable name	Description	Example or Default
ssl_auth_spsra	SSL authorization to use for network connection.	1 (Default)
port_spsra	Port to use for network connection. By default, uses the port defined for the targetable component.	:[targetableComponent:sps_ra_port] (Default)
adv_params_spsra	Advanced parameters to use for network connection. By default, uses the advanced parameters defined for the targetable component.	:[targetableComponent:sps_ra_adv_parms] (Default)

General OS Provisioning Plug-In Variables

The variables listed in the following table are general variables that the OS provisioning plug-in requires. They are not technically part of the JET modules, but are required for the OS provisioning plug-in to interface between JET and the N1 Grid SPS software.

TABLE B–3 General OS Provisioning Component Variables

Variable name	Description	Example or Default
installPath	Jumpstart Enterprise Toolkit (JET) virtual host	myhost-jet (Example)
osp_target_host	Target host name	:[targetableComponent:installPath] (Default)
osp_profile_control_service	OS provisioning control server (TRUE, FALSE)	:[targetableComponent: osp_control_service] (Default)
sps_ra_parent_host	Host name of the N1 Grid SPS RA parent	:[targetableComponent:sps_ra_parent] (Default)
sps_ra_parent_host_type	Host type of the N1 Grid SPS RA parent (MS or LD)	:[targetableComponent: sps_ra_parent_type] (Default)
osp_jet_server	Host name of the JET server	myserver (Example)
osp_profile_architecture	Target host architecture (sun4u, x86)	:[targetableComponent:architecture] (Default)

TABLE B–3 General OS Provisioning Component Variables *(Continued)*

Variable name	Description	Example or Default
`osp_profile_min_disk_size`	Minimum disk size	`5GB` (Example)
`osp_profile_description`	Profile description	
`osp_profile_monitor_install`	OS provisioning monitor (TRUE, FALSE)	`FALSE` (Default)
`osp_profile_boot_server_name`	Solaris boot server name	`:[osp_jet_server]` (Default)
`osp_profile_BS_subnet_address`	Subnet served by boot server	
`osp_profile_BS_subnet_ip_address`	IP address of boot server on this subnet	`199.199.8.6` (Example)
`osp_profile_product_ip_address`	IP address of Solaris image server	`199.199.8.6` (Example)
`osp_profile_product_directory`	Path to image directory on image server	`/osp_image/Solaris9_u7_sparc` (Example)
`osp_profile_boot_ip_address`	IP address of Solaris boot server	`199.199.8.6` (Example)
`osp_profile_boot_directory`	Solaris boot directory	`/osp_image/Solaris9_u7_sparc/Solaris_9/Tools/Boot` (Default)
`osp_profile_sysidcfg_ip_address`	IP address of system ID configuration	`199.199.8.6` (Example)
`osp_profile_sysidcfg_directory`	Directory of system ID configuration	`/opt/SUNWjet/Clients/:[osp_target_host]` (Default)
`osp_profile_jumpstart_ip_address`	Jumpstart IP address	`199.199.8.6` (Example)
`osp_profile_jumpstart_directory`	Jumpstart directory	`/opt/SUNWjet` (Default)
`osp_profile_boot_file_name`	Solaris boot file name	`:[osp_target_host]` (Default)
`osp_profile_nbp_boot_file_name`	Network boot file name	`:[osp_target_host]` (Default)

APPENDIX C

Creating Custom JET Modules

Scope

The module is the glue between the JET framework and the facilities provided to perform the actual installation of the product software. The module presents a set of configuration options in the template to allow per-target server options to be set, which are then used to drive the product installation according to those parameters.

There are no special requirements or restrictions about the scope of what options are presented in the template, or how to drive the product installation. There are some guidelines however, to try to ensure that modules co-exist with each other as much as possible and have no strange dependencies etc.

Module Design Guidelines

The toolkit itself was designed around some basic principles. The modules that were initially created alongside the toolkit also followed this style. While every module developer has the option of going down their own path, the following should be at least considered and followed if possible.

Module Coverage

Each module should cover a particular scope of an application, preferably without replicating the functionality available in another module. In addition, a module should be appropriately sized; if the module can be sensibly broken into sub components, then smaller modules should be created instead of one large monolithic module.

For example: A designer may decide to write a module to provide security services to a target server installation. As part of this module, the designer decides to install a firewall and a set of hardening scripts.

In this case, the designer should investigate whether another module already covers either of these two areas and could be leveraged. In addition, if either the firewall or hardening scripts may be used elsewhere in isolation from each other, it may be more flexible to create them as two modules rather than bundling them together.

Module Dependencies

Each module should be self sufficient and not expect or rely on another module to be present. If such interaction is required (and sometimes it does make sense), then the module should be clearly identified as being dependent on the other module. If the situation does arise, it should be determined whether the functionality is best represented in distinct modules or whether the two modules that are dependent on each other are better represented in one encompassing module. The basic premise is to keep items together where it makes sense and not to create lots of smaller modules just for the sake of it.

Module Interactions

Where the case is strong for module separation and the modules are dependent (in at least one direction), the modules should try to interact with each other so that the correct (desirable) outcome is achieved. The toolkit has a simplistic mechanism for setting and retrieving hints on a per module basis. Whether a module will pick up the hints, is up to the designer of the module, but with cooperation between module developers, the modules can be written in such a way that they work happily in isolation, but when put together they provide a stronger solution.

For example: From a real-world example, the Sun Cluster 3 product places dependencies on how Solstice DiskSuite is configured, but does not always need DiskSuite installed, nor does DiskSuite require the cluster software installed indeed it is very valuable to be installed on non-clustered machines.

In this case, we have two very distinct modules; one that covers DiskSuite and one that covers the Sun Cluster product. Each works fine in isolation, but when they are combined within the same target server configuration, the Sun Cluster module influences the DiskSuite module so it conforms to the restrictions placed on DiskSuite by Sun Cluster 3.0.

This interaction is done through the use of module hints and their behavior in the standalone scenarios can be summarized as follows:

- When only the Sun Cluster module is active, Sun Cluster sets the hints, but DiskSuite is not present to pick them up

- When only DiskSuite is active, it looks for the hints, but they have not been set by any other module, so it uses it's regular default behavior.

What the hints actually represent is totally up to the module developer. Close cooperation between module developers will enable the most efficient use of hints; if possible, the hints should be documented within the module Release Notes, so other module developers may take advantage of the additional interfaces.

Module Coding

The module developer should try to select a standard scripting language that will be available during the JumpStart installation. For example, the use of a 'bash' is not possible, as the NFS boot image the target server uses does not contain that shell.

Where possible, the Bourne Shell should be used as this is know to exist on all versions of Solaris. Only as a last resort should a compiled language be considered.

Module Directory

Every module has its own directory structure, located in the `Products` subdirectory from where the main toolkit framework was installed. The name of the directory is the name used by the toolkit to reference the particular module.

For example, the module `sds` (Solstice DiskSuite) resides in `.../Products/sds/`, and it alone controls what resides in that directory and how it is used; with some notable exceptions listed below.

No further interaction is required to register a module with the toolkit ; the presence of the directory is sufficient. It is not advisable to create symbolic links in the `Products` directory to other locations outside of the main toolkit installation point; the target servers may not be able to access such directories during the installation, unless other measures have been taken to provide such functionality.

Module Configuration (*module*`.conf` File)

Each module is intended to be configurable to some extent by the user; although this is not a mandatory requirement, it is normally expected to be implemented.

The toolkit takes a very simplistic view on providing module configuration to the user. When use the toolkit `make_template` command to create a template, a single flat file is produced. This file concatenates the core base_config configuration file with the configuration files from each of the selected modules. By providing a configuration file named after the module itself and with a `.conf` suffix, the toolkit `make_template` command will do the rest of the work.

The configuration file is a simple Bourne Shell script. The configuration file should present configurable options in the form of variables to the user. Each variable should be prefixed with the module name and an underscore, to preserve the variable namespace of the module and prevent one module corrupting another.

For example, the module `sds` presents an option to the user to select the version of the software to be installed. The corresponding part of the configuration file is as follows:

```
############
#
# Which version of the product is to be installed
#
sds_product_version="4.2.1"
```

In this example, a default value of 4.2.1 is already populated in the configuration file, as this was the most recent version of the product when the module was written.

Module Interfaces

The toolkit calls particular interfaces within the module during the life cycle of the JumpStart process - both on the target server and the JumpStart server. Each interface is expected to be an executable shell script (or worst case, a binary), with the context of the target server supplied through environmental variables.

copy_media Interface

Called on	JumpStart server
Arguments	`<patches|packages>` *version srcdir destdir arch*
Required or optional?	Required

The copy_media script is called when the user calls the `copy_product_media` or `copy_product_patches` scripts to manage the media for this module. The script should understand what format the application is delivered in, and perform the copy from the passed source media location, to the appropriate media location on the server. This function allows the module to handle exotic forms of media (tar, gz, zip, bz2, and so on), without the main toolkit requiring to be updated for each media type. It also enables the module developer to place the media onto the server in a known state. For example, a product may require a whole directory tree to be visible, or just a bunch of Solaris packages.

make_template Interface

Called on	JumpStart server
Arguments	none
Required or optional?	Optional

When an administrator wants to create a new definition for a server build, they will execute the top level command `make_template` supplied in `/opt/SUNWjet/bin`. This top level `make_template` script will set up the basic target server configuration information and then call each module specific `make_template` script, if it is present. The module specific `make_template` script can perform additional work on the template; for example, populating per-client defaults for the user to edit.

make_client Interface

Called on	JumpStart server
Arguments	none
Required or optional?	Optional

When an administrator wants to set up a target server for installation, they will execute the top level command `make_client` supplied in the bin directory of the toolkit. This top level `make_client` script will set up the basic target server configuration information and then call each module specific `make_client` script if it is present. The module specific `make_client` script can perform additional work in the target server specific `/opt/SUNWjet/Clients` directory, it may configure module hints, modify the target server profile, `sysidcfg` or other files accordingly.

begin Interface

Called on	target server
Arguments	none
Required or optional?	Optional

During the "begin" phase of the JumpStart process, the toolkit will check to see if the module has a script called `begin`, and if it is present, the script will be executed. Any variables set in the module configuration section of the template will be present in the environment for the script to access.

install Interface

Called on	target server
Arguments	none
Required or optional?	Required

The install script is the main workhorse of the module. It is called on the target server after the main Solaris installation has completed, during the "finish" script phase. Its purpose is to instigate the installation or configuration of the particular application.

The script itself is called prior to the first reboot of the newly installed target server. At this point, the root directory (`/`) is actually an NFS mounted filesystem from the JumpStart server. The real disk based root directory is located through the use of the environment variable $ROOTDIR, traditionally set to `/a`.

If the application cannot be installed when root is located on $ROOTDIR, the install script should use the post-installation functionality provided by the toolkit, to schedule a subsequent installation after the first reboot. After the first reboot, the target server will actually boot of its own disk, and root will really be on `/`.

The install script is responsible for taking the user supplied configuration and driving the actual application installation and configuration appropriately. How this is achieved, is up to the ingenuity of the module developer, although a number of utility functions are available from the main toolkit, to assist with the common tasks; such as package or patch installation, file copies, message reporting etc.

Prior to the module install script being called, the module configuration as defined in the template and originally populated from the *module*`.conf` file is loaded into the shell environment. The install script should not expect to be called with arguments, but should instead take its configuration from the current environment. This technique avoids the problem of each module installation script requiring different numbers of arguments etc.

check_client Interface

Called on	target server
Arguments	none

Required or optional?   Optional

The check_client script can be employed by a module developer to perform basic checks about the configuration options specified in the template. When called, the environment will be configured with the variables set in the template, and the script can perform basic checking to try to reduce any installation errors.

The module may decide to check for valid options or that media exists for the selected version etc. The level of functionality provided is up to the implementor.

Support Functions in the Toolkit

The main toolkit supplies many common functions that can be utilized by modules. This allows for better code re-use and simpler modules. The best way to find out what functions are available, is to examine the directory `/opt/SUNWjet/Utils/lib` in which they reside.

Adding a JET Module

This section describes how you can add additional JET modules to the Solaris boot and install (JET) server. Although the process is describes in the context of a Flash module, the logic extends to other JET modules.

▼ How to Add a Flash Module

Steps

1. **Download the JetFLASH.pkg onto the Solaris boot and install server.**

2. **Install the JetFLASH package.**

   Use commands similar to the following example:

```
# cat >/tmp/admin-file <<- _EOF
mail=\n
instance=unique
partial=quit
runlevel=ask
idepend=quit
rdepend=nocheck
```

```
space=quit
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=/opt/SUNWjet/Products
_EOF
  pkgadd -a /tmp/admin-file -d JetFLASH.pkg
```

3. **Copy the Flash image to the Solaris boot and install server.**

   Use commands similar to the following example:

   ```
   # telnet solaris-bis-ip-address
     # cp flash-archive /export/install/flash/sol10_xall_sparc.flar
   ```

4. **Create a Solaris Profile for this Flash image.**

   Use commands similar to the following example:

```
# cat > /tmp/flash-profile <<- _EOF
flash-with-ra
Solaris10 Flash Archive With SPS RA
base_config flash spsra
_EOF
cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/n1osp/untyped/SolarisImage-attach \
-tar H:NM:masterserver-jet -comp + -vs solaris10 -pto 120 -nto 120 -f /tmp/flash-profile
```

Tip – You can also create the Profile component through the N1 Grid SPS browser interface. Use the Solaris Image: create profile option.

5. **Edit the Profile component that you created in the previous step.**

   The profile is located at `/com/sun/n1osp/autogen-masterserver-jet/provision/`. In this example task, the Profile component is `/com/sun/n1osp/autogen-masterserver-jet/provision/Solaris_10.flash`.

6. **In the Profile component, change the `archive_locations_flash` variable value to point to the flash archive.**

   For example:

```
archive_locations_flash nfs://10.216.0.55/export/install/flash/sol10_xall_sparc.flar
```

7. **Check in the component.**

   You are now ready to use the above Solaris Profile to deploy on desired target host.

Index